INPUT 64

DAS ELEKTRONISCHE MAGAZIN 9/85

Infos · News · Programme · Unterhaltung · Tips

Bundesliga-Manager
Kampf um die Tabellenspitze

ReAss
Maschinenprogramme entschlüsseln

Für Geschäft und Geburtstage:
Terminkalender

Spiele, Serien, Kurse

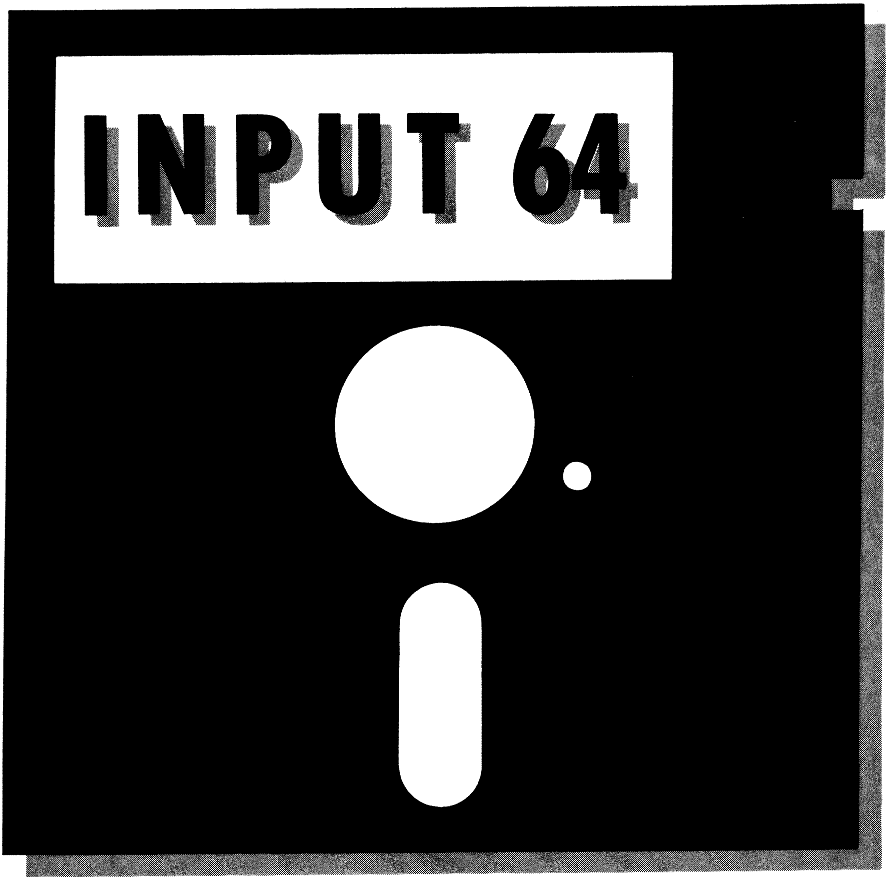
Im Test:
80-Zeichen-Karten

Hilfsprogramme:
Sound-Tools



Dokumentation und Bedienungshinweise

INPUT-ein Magazin aus dem Verlag Heinz Heise GmbH, Postfach 61 04 07, 3000 Hannover 61


INPUT 64

INHALTSVERZEICHNIS INPUT 64

9/85

Bedienungsanleitung	Seite 2
Leser fragen. . .	Seite 4
Terminkalender	Seite 6
Superkäsi	Seite 11
Mathe mit Nico	Seite 12
Wettbewerbsgewinner: ReAss	Seite 12
News: Was bringen 80-Zeichen-Karten?	Seite 22
3000.-DM-Wettbewerb	Seite 24
64er-Tips	Seite 28
Cave Climber	Seite 29
Manager	Seite 30
SID-Kurs	Seite 33
Hinweise für Autoren	Seite 34
Hilfsprogramme: Sound-Erweiterung	Seite 35
Für Nachzügler und "Spätzünder"	Seite 40
Rätselecke	Seite 41
Bei Ladeproblemen	Seite 45
Vorschau	Seite 47

Liebe 64er-Besitzer(in),

wenn ein Teil unserer Leser uns schreibt, er interessiere sich vor allem für die Kurse in INPUT 64, andere meinen, sie kauften das Magazin nur wegen der Hilfsprogramme, und wieder andere die Aussicht auf neue Spiele jeden Monat zum Kiosk treibt - dann schliessen wir erst einmal daraus, daß wir mit unserem Konzept im Prinzip richtig liegen.

Andererseits bleiben natürlich viele Fragen offen. Vor allem: Gibt es vielleicht einen Bereich, der Sie interessiert, aber innerhalb des Magazins gar nicht vorkommt? Mit anderen Worten: was fehlt Ihnen in INPUT 64?

Schreiben Sie uns doch mal!

Ihre INPUT 64-Redaktion

Hinweise zur Bedienung

1.) Entfernen Sie - bei ausgeschaltetem Rechner - eventuell vorhandene Steckmodule. Schalten Sie vor dem Laden von INPUT 64 ihren 64er einmal kurz aus und dann wieder ein.

2.) Geben Sie zum Laden der Kassette (LOAD) und (RETURN) ein. Diskettenbesitzer: (LOAD"INPUT*",8,1) und (RETURN). Alles weitere geschieht von selbst. Nach der Titelgrafik springt das Programm ins Inhaltsverzeichnis des Magazins.

3.) Das können Sie nun mit der Leertaste durchblättern und mit (RETURN) das angezeigte Programm auswählen. Im Fenster unten rechts erhalten Sie dann weitere Hinweise ("Bitte Band zurückspulen" und so weiter ...).

4.) Merken Sie sich fünf Kommandos:

- (CTRL) und gleichzeitig (h) ruft aus dem laufenden Programm eine Hilfsseite des INPUT 64-Betriebssystems auf. Sie finden darauf weitere Hinweise. Drücken Sie erneut (CTRL) und (h), verschwindet das eingeblendete Fenster, und es geht weiter im Programm.

- (CTRL) und (i) beendet das aktuell laufende Programm und führt zurück ins Inhaltsverzeichnis. Sie können dann ein anderes Modul anwählen oder das bereits geladene Programm neu starten. (Letzteres

funktioniert nicht immer, einige Programme lassen dies nicht zu. Sie werden dann zum "Band zurückspulen" aufgefordert. Von Diskette wird automatisch neu geladen.)

- (CTRL) und (h) bzw. (CTRL) und (i) können Sie im Prinzip immer anwählen.

- (CTRL) und (s) steht für SAVEn und funktioniert bei allen Programmen, die auch außerhalb von INPUT 64 anwendbar sind: Spiele, Tools, Anwenderprogramme etc. Legen Sie Ihre eigene Kassette oder Diskette ein, drücken Sie (CTRL) und (s) und befolgen Sie die weiteren Anweisungen - Sie haben eine Kopie des Programms. Diese Option ist bei vielen Programmen nur am Programmanfang möglich - entnehmen Sie dies jeweils den Hinweisen zu den einzelnen Modulen in diesem Heft.

- (CTRL) und (q) ist für die ganz Eiligen: Mit diesen beiden Tasten können Sie das Titelbild abkürzen.

- (CTRL) und (b) ermöglicht einen Bildschirmausdruck - natürlich nicht von Grafikseiten oder Sprites! Angepaßt ist diese Hardcopy für Commodore-Drucker und kompatible Geräte. Denken Sie bitte daran, immer zuerst den Drucker und dann den Rechner einzuschalten. Voraussetzung ist Geräteadresse 4 oder 6.

6.) Haben Sie bei der Auswahl eines Programms eventuell nicht weit genug zurückgespult, und es wurde nicht gefunden, spulen Sie bis zum Bandanfang zurück. Diskettenbesitzer stellen bitte sicher, daß noch die INPUT 64-Diskette eingelegt ist.

Auf der Rückseite der Verpackung finden Sie das Inhaltsverzeichnis der Kassette/Diskette.

Auf einen Blick: INPUT 64-Betriebssystembefehle

Inhaltsverzeichnis aufrufen	CTRL und i
Hilfsseite aufrufen	CTRL und h
Programm abspeichern	CTRL und s
Bildschirmausdruck	CTRL und b
Titelbild abkürzen	CTRL und q

Auf der 2. Kassettenseite befindet sich eine Sicherheitskopie von Seite 1. Sollten Sie eventuell mit einem der Programme auf der 1. Seite Ladeschwierigkeiten haben, versuchen Sie es auf Seite 2. Führt auch dies nicht zum Erfolg, lesen Sie bitte die entsprechenden Hinweise im Kapitel "Bei Ladeproblemen"!

PS: Drücken Sie nicht RUN/STOP und RESTORE. Dadurch kann der Rechner "abstürzen". Gelangen Sie doch versehentlich in den Direktmodus, so befördert Sie ein SYS50307 zurück in das INPUT 64-Betriebssystem.

Leser fragen...

Ohne Frage

Bei Ihrem Supertape DII (Ausgabe 4/85) stört mich die Abfrage "Maschinenprogramm sichern", die nach jedem Start des Programms erscheint. Kann man das einfach ändern?

(Hartlage, Langenhagen)

Man kann. Wenn Sie nach dem Laden des Programms eingeben
POKE 2231,208 : POKE 2232,99
und das Programm dann wieder abspeichern, wird es sich nach dem Start künftig sofort mit "SUPERTAPE AKTIVIERT" melden. Diese Veränderung kann übrigens nur an der in Ausgabe 4,85 veröffentlichten Version durchgeführt werden. Durch diese Änderung wird vor der Abfrage ein Sprungbefehl (branch by not equal) eingefügt, dessen Bedingung an dieser Stelle des Programms immer zutrifft, und es geht direkt zur Initialisierung.

(d.Red.)

Spät entdeckt

Der Zeichensatzgenerator (Ausgabe 2/85) spielte bislang ganz gut. Als ich allerdings die reversen Sonderzeichen ändern wollte, waren die neuen Zeichen nach dem Abspeichern und Neuladen des Zeichensatzes verschwunden!

(kein Brief, sondern Anmerkung eines Autors)

Bestimmte Bugs werden entscheidend erst Monate später entdeckt. Folgender POKE behebt das Problem:
POKE 4583,26 (Dadurch wird auch der letzte Block des erzeugten BASIC-Files mit abgespeichert und nicht, wie bislang, 32 Zeichen zu wenig.)
Also: Zeichengenerator laden (nicht starten!), POKE durchführen, korrigierte Version abspeichern.

(d. Red.)

Anpassung von MOTORBIKE an Autos

" ... in INPUT 5/85 stellen Sie das Programm "MOTORBIKE" vor. Da dieses Programm für mich sehr interessant ist ... wäre ich Ihnen sehr dankbar, wenn Sie mit mitteilen würden, wie dieses Programm für PKW umgeschrieben werden kann ... "

(S. Biniasz, Bremen)

Die Anwendung der in Input 5/85 versprochenen Tips zur Anpassung von Motorbike an Autos setzt voraus, daß Sie über eine umfangreiche technische Dokumentation, zum Beispiel "Querschnitt durch die Automobiltechnik", zu Ihrem Wagen verfügen.

Anpassung des Motors: Motorbike geht bei der Dimensionierung der Ansaugwege und Vergaser von je einem Vergaser pro Zylinder aus und optimiert auf guten Durchzug. Da bei Automotoren oftmals nur ein Einzelvergaser für 4 Zylinder vorhanden ist, muß dieser natürlich im Durchlaß größer sein. Dazu wird der Wert K3=.675 in Zeile 6100 auf K3=.82 geaendert.

Anpassung der Uebersetzung: Hier liegen die Dinge leider nicht so einfach wie beim Motor. Ersetzen Sie bitte in Zeile 10520 die Werte für "gf" und "gm" durch ihre eigenen Werte (siehe auch das Begleitheft 5/85, Seite 26). Analog zu Zeile 10520 ändern Sie Zeile 10200 (Reifen). Falls Ihr Auto ein 5-Gang-Getriebe hat, können Sie die Einteilung von Zeile 10210 übernehmen. Bei einem 4-Gang-Getriebe fällt "g5" weg. Passen Sie die Zeilen 10260,10740 und 10750 entsprechend an, und achten Sie darauf, daß kein "Division by zero error" auftritt.

Die Werte für die Gangstufen sind die Werte der Primaerübersetzung, also des Getriebes. Die Werte für Ritzel und Kettenblatt entsprechen den Werten für die Zahnräder möglicher Hinterachsen. Eine Routine zur Berechnung der Sekundaerübersetzung, also dem Produkt von Ritzel und Kettenblatt, ist in Motorbike nicht vorgesehen, Sie müßten das Programm entsprechend umschreiben.

Anpassung der Fahrleistungen: Falls Sie das Getriebe erfolgreich angepaßt haben, sollte Motorbike auch die Beschleunigung 0-100 km/h korrekt berechnen. Leider kann die Höchstgeschwindigkeit nicht wie bei Motorrädern mit einer Faustformel hinreichend genau ermittelt werden.
Sie müssen hier von der folgenden Formel ausgehen:

Höchstgeschw.= Faktor * dritte Wurzel aus Nennleistung

Der Faktor besteht aus dem cw-Wert * der Stirnfläche des Autos. Falls Sie Stirnfläche oder cw-Wert nicht kennen, muß der Faktor über einen "Trick" ermittelt werden.
Die Formel wird nach dem Faktor (F) umgestellt:

$$F = \frac{\text{Höchstgeschwindigkeit.}}{\text{3. Wurzel aus der Nennl.}}$$

Die Höchstgeschwindigkeit kann man Testberichten oder dem Kfz-Schein entnehmen. Die Nennleistung entspricht dann genau der Serie. Ersetzen Sie die "45" durch den so ermittelten Faktor in Zeile 10170, um die Höchstgeschwindigkeit des getunten Fahrzeugs zu ermitteln. Den Wert für den liegenden Fahrer (vl) bzw. die Verkleidung (vf) müssen Sie streichen. (Division by zero error vermeiden und die Bildschirmausgabe überarbeiten!).

H.-J. Krieghoff

Terminkalender — ein PASCAL-Programm —

Ihre Termine werden vom C 64 verwaltet. Sie können bis zu 250 Termine im Speicher anlegen und anschließend auf Datenträger abspeichern. Da Sie verschiedene Termin-Dateien anlegen können, ist die Kapazität nahezu unbegrenzt.

Ihre Termine können frühestens 1985 beginnen und auch über mehrer Jahre verteilt werden. Solche Dauertermine/Jahr bleiben im Speicher, bis Sie von Ihnen gelöscht werden.

Das Programm wurde in PASCAL geschrieben. Auf den folgenden Seiten sehen Sie das Listing.

Die Bedienungs-Anleitung finden Sie auf Seite 26.

Listing Termin-Kalender

```
program terminkalender;
;
(* variablen + konstanten *)
;
const max=200;
var gefunden:packed array[0..207] of boolean;

;    floppy,8,5:file;
;    cassin,1,0:file;
;    cassout,1,1:file;
;    turboin,7,0:file;
;    turboout,7,1:file;
;    drucker,4,7:file;
;    z1,z2,device,null,menuewahl,druckstop:char;
;    dateiname:array[1..14] of char;
;    string,bottomstring:array[1..40] of char;
;    floppystring:array[1..21] of char;
;    tagdatum,suchname,datum:array[1..10] of char;
;    tag,monat,jahr,code,t,m,j,c,st,sm,sj,sc,lauf,i1,i2:integer;
;    eingabeart,tagnummer:integer;
;    tage:array[1..12] of integer;
;    tagname:array[1..7,1..10] of char;
;    termin:array[1..max] of record
;           a:integer;
;           b:integer;
;           c:array[1..37] of char;
;           end; (* of record *)
;
(* funktionen + proceduren *)
;
;  function eof (dummy:integer):boolean;
;      begin
;         eof:= (peek(144) and 64)=64;
;  end; (* of function *)
;
;  procedure formatausgabe;
;     begin
;       string:= "|                                      |";
;       string[2..0]:= termin.c[lauf,
1..length(termin.c[lauf])];
```

```
;  end; (* of procedure *)
;
;  procedure stringinput (laenge:integer);
;    var z:char;
;        j:integer;
;    procedure zurueck;
;      begin
;      poke 204,1;
;      j:= j-1; write ("   ");
;      string[j]:= chr(0); poke 204,0;
;      end; (* of proc *)
;    begin integer;
;      j:= 1; z:= chr(0);
;      if laenge<>10 then fillchar (string,chr(0));
;      poke 204,0;
;      while z<>chr(13) do begin
;        while not((j>laenge) or (z=chr(13))) do begin
;          repeat get z;
;          if (z=chr(20)) and (j>1) then zurueck;
;   until (z>=" ") and (z<chr(96)) or (z>="A") and (z<="Z") or (z=chr(13));
;          if z<>chr(13) then begin
;             string[j]:= z;
;             j:= j+1; write (z);
;          end;
;        end;
;        if z<>chr(13) then get z;
;        if z=chr(20) then zurueck;
;      end;
;      poke 204,1; writeln (" ");
;  end; (* of procedure *)
;
;  procedure schaltjahr (jahr:integer);
;      begin
;      if (((jahr mod 4=0) and (jahr mod 100<>0)) or (jahr mod 400=0)) then
;        tage[2]:= 29;
;      else tage[2]:= 28;
;  end; (* of procedure *)
;
;  procedure geraet;
;      begin
;      writeln ("Arbeiten Sie mit:");
;      writeln ("Cassette = 1");
;      writeln ("Supertape= 7");
;      writeln ("Diskette = 8");
;      writeln ("Eingabe der Geraetenummer! ");
repeat get device;
until (device="1") or (device="8") or ((device="7") and (peek(817)<>244));
;  end; (* of procedure *)
;
;  procedure datenart;
;    var z:char;
;      begin write ("",chr(142));
;        write (" U────────");
;        write ("─────────────────────────");
;        write ("──I  | terminkalender **");
;        write (" eingabe/loeschen |  J──");
;        write ("─────────────────────────");
;        writeln ("──────────K");
;        write ("  ┌────────");
;        write ("────────────────┬────────");
;        write ("──┐  |                 ");
;        write ("     |          |  | 0 ");
;        write (": hauptmenue         |bitt");
;        write ("e          ├─────────────");
;        write ("──────────┤           | ");
;        write (" | 1 : normaltermine      ");
;
;          write (" |Funktionsart  ├──────────");
;          write ("──────────────────┤          ");
;          write ("    |  | 2 : dauertermine/");
;          write ("jahr  |mit ziffer   |  ├──");
;          write ("──────────────────────┤    ");
;          write ("          |  | 3 : dauerter");
;          write ("m./woche  |anwaehlen !  | ");
;          write (" |                        ");
;          write ("|               |  └──────");
;          write ("──────────────────┴───────");
;          writeln ("───┘");
;          repeat get z;
;          until (z>="0") and (z<"4");
;          eingabeart:= ord(z)-48;
;          writeln ("");
;  end; (* of procedure *)
;
;  function sicher (dummy:integer):char;
;      begin
;        writeln ("aktueller dateiname: ",dateiname);
;        writeln ("sind sie sicher ? (j/n)");
;        repeat get sicher;
;        until (sicher="j") or (sicher="n");
;  end; (* of function *)
;
;  function ioerror (dummy:integer):boolean;
;    var irrtum:array[1..32] of char;
;    begin integer;
;    poke 144,(peek(144) and 191);
;    i1:= 1; fillchar (irrtum,chr(0));
;    while not(eof(1)) do begin read (error,irrtum[i1]); i1:= i1+1; end;
;    if irrtum[1..2]<>"00" then begin          writeln ("",irrtum);
;      ioerror:= true; end ;
;    else ioerror:= false;
;    for i1:= 1 to 15000 do begin end;
;    poke 144,(peek(144) and 191);
;  end; (* of function *)
;
;  procedure initial (l1:integer);
;    begin
;      termin.a[l1]:= 0;
;      termin.b[l1]:= 0;
;      fillchar (termin.c[l1],chr(0));
;  end; (* of procedure *)
;
;  procedure halt;
;      begin
;         if peek(214)>20 then
;         repeat until peek(197)<>64;
;         else exit;
;         write ("");
;         poke 198,0;
;  end; (* of procedure *)
;
;  procedure tageingabe;
;    begin
;      repeat
;       tagnummer:= -1;
;       write ("Name d.","Wochentages -> ");
;       stringinput (11); writeln ("");
;       suchname:= string[1..10];
;       for i1:= 1 to 7 do
;         if suchname=tagname[i1] then tagnummer:= i1;
;      until tagnummer<>-1;
;  end; (* of procedure *)
;
```

```
; function schluss (art:integer):char;
;   begin
;      if art=1 then begin
;        poke 214,23; writeln;
;        write ("■ Schluss ? (j/n)  ■");
;        repeat get schluss;
;        until (schluss="j") or (schluss="n");
;      writeln ("■");
;      end; (* of if *)
;      else schluss:= "j";
; end; (* of function *)
;
; procedure eingabe;
;     begin
;        write ("■■┌──────────────────────────────────────┐");
;        write ("| Eingabe des"," Termins :             |");
;        write ("├","──────────────────────────────────────","┤");
;        write ("|                                      |");
;        write ("└──────────────────────────────────────┘■■■");
;        stringinput (37);
;        termin.c[lauf]:= string[1..37];
; end; (* of procedure *)
;
;
; procedure weiter;
;     begin
;     integer;
;     schaltjahr (sj);
;     if st<tage[sm] then st:= st+1;
;     else begin
;        if sm<12 then begin
;           sm:= sm+1; st:= 1;
;           end; (* of if *)
;        else begin
;           sj:= sj+1; sm:= 1; st:= 1;
;           end; (* of else *)
;        end; (* of else *)
;        sc:= 32*sm+st;
; end; (* of procedure *)
;
; procedure wochentag (ta,mo,ja:integer; var index:integer);
;     var k,l,o,z1,z2,z3,z4,z:integer;
;         p:real;
;     begin
;     integer;
;        k:= trunc(0.6+1/mo);
;        l:= ja-k;
;        o:= mo+12*k;
;        p:= 1/100;
;        z1:= p div 4;
;        z2:= trunc(p);
;        z3:= trunc(5*1/4);
;        z4:= trunc(13*(o+1)/5);
;        index:= z4+z3-z2+z1+ta-1;
;        index:= index mod 7+1;
; end; (* of procedure *)
;
; procedure dateneingabe (var t2,m2,j2,c2:integer);
;   begin integer;
;        repeat
;        string[1..10]:= tagdatum;
;        writeln ("■■■■┌────────────────────┐");
;        writeln ("■■| Datum eingeben!   |");
;        writeln ("■■├","────────────────────","┤");
;        writeln ("■■|   ",tagdatum,"       |");
;        writeln ("■■└────────────────────┘");
```

```
write ("CCCCCCCCCCCCCCCCCCCCCCCCCC");
write ("CCCCCCCCK                  ");
write ("                           ");
write (" ■JCCCCCCCCCCCCCCCCCCC┬CCCC");
write ("CCCCCCCCCCCCCI   |■■ 1 ■ abfr");
write ("age          ■|■■ 2 ■ neueingabe");
write ("    ■|   ├","CCCCCCCCCCCCCCCCCCC");
write ("┼","CCCCCCCCCCCCCCCCC┤   |■■ 3 ");
write ("■ loeschen        ■|■■ 4 ■ druck");
write ("en        ■|   ├","CCCCCCCCCCCCC");
write ("CCCCC","┼","CCCCCCCCCCCCCCCCC┤ ");
write (" |■■ 5 ■ datei laden    ■|■■ 6 ■ ");
write ("dateiname     ■|   ├","CCCCCCCC");
write ("CCCCCCCCCC","┼","CCCCCCCCCCCCCC");
write ("CCC┤   |■■ 7 ■ datei saven    ");
write ("■|■■ 8 ■ neustart      ■|   KCCC");
write ("CCCCCCCCCCCCCCC","┼","CCCCCCCC");
write ("CCCCCCCCK   |■■ @ ■ deviceadr");
write ("esse ■|                        ");
write (" JCCCCCCCCCCCCCCCCCCK■■       ");
write ("                      ┌CCCCCCCC");
write ("CCCCCCCCCCCCCCCCCC┬CCCCCC");
write ("CCC┐   | ■■dateiname:■ ",dateiname);
write ("■■| ■■dev.:■■ ");
write (device,"  |   ├","CCC");
write ("CCCCCCCCCCC","┬","CCCCCCCCCC","┴","C");
write ("CCCCCCCC┤   | ■■tagesdatum:■ ");
write (" ■■|■alle termine in der■■   | ");
write (" |                    |■Form   <");
write ("tt.mm.jjjj>■■   |   | ",tagdatum);
write ("     |■eingeben ! --------");
write ("-■■   |   └CCCCCCCCCCCCC","┴","CCC");
writeln ("CCCCCCCCCCCCCCCCCC┘");
;  repeat get menuewahl;
;  until ((menuewahl>="1") and (menuewahl<="8")) or (menuewahl="@");
;
;     case menuewahl of
;
;       "@": geraet;
;       "1": begin writeln ("■■■");
;            dateneingabe (st,sm,sj,sc);
;            repeat
;              wochentag (st,sm,sj,tagnummer);
;             writeln ("■■■■* ",st,".",sm,".",sj,": ",tagname[tagnummer]);
;              writeln ("■────────────────────────────────────────");
;              for lauf:= 1 to max do begin
;                  if test(lauf) then begin
;                    writeln (" ┌──────────────────────────────────────┐");
;                    formatausgabe; writeln (string);
;                    writeln (bottomstring);
;                  end; (* of if *)
;                  halt;
;              end; (* of for *)
;              weiter;
;            until schluss(1)="j";
;            end; (* of case1 *)
;       "2": repeat i2:= 1;
;            lauf:= 1; while termin.a[lauf]<>0 do lauf:= lauf+1;
;            if lauf<=max then begin
;              datenart;
;              case eingabeart of
;                  0: i2:= 2;
;                1,2: begin
;                       dateneingabe (tag,monat,jahr,code);
;                       termin.a[lauf]:= code;
;                       if eingabeart=1 then termin.b[lauf]:= jahr;
```

```
;          write ("[illegible]");
;          stringinput (10);
;          datum:= string[1..10]; writeln ("[illegible]",datum,"[illegible]");
;          t2:= val(datum[1..2]);
;          m2:= val(datum[4..5]);
;          j2:= val(datum[7..10]);
;          c2:= m2*32+t2;
;          schaltjahr (j2);
;          if (m2<1) or (m2>12) then c2:= 0;
;          else if (j2<1985) or (j2>2100) then c2:= 0;
;            else if (t2<1) or (t2>tage[m2]) then c2:= 0;
;         until c2<>0;
;  end; (* of procedure *)
;
;  function test (l1:integer):boolean;
;      begin integer; test:= false;
;      if (termin.a[l1]=sc) and ((termin.b[l1]=0) or (termin.b[l1]=sj)) then
;             test:= true;
;      if termin.b[l1]=tagnummer then test:= true;
;  end; (* of function *)
;
;  procedure datloeschen;
;     function loeschen (dummy:integer):char;
;       begin
;         writeln ("[illegible]        L o e s c h e n  ?  (j/n)        [illegible]");
;         repeat get loeschen;
;         until (loeschen="j") or (loeschen="n");
;     end; (* of function *)
;     begin
;      writeln ("[illegible]",termin.c[lauf]);
;      if loeschen(1)="j" then initial (lauf);
;  end; (* of procedure *)
;
(* ---------------------------- *)
(* hauptprogramm *)
;
begin
;
start:
;  integer;
;
;  tagname[1]:= "Sonntag"; tagname[2]:=  "Montag";
;  tagname[3]:= "Dienstag"; tagname[4]:= "Mittwoch";
;  tagname[5]:= "Donnerstag"; tagname[6]:= "Freitag";
;  tagname[7]:= "Samstag";
;  tage[1]:= 31; tage[2]:= 28; tage[3]:= 31; tage[4]:= 30; tage[5]:= 31;
;  tage[6]:= 30; tage[7]:= 31; tage[8]:= 31; tage[9]:= 30; tage[10]:= 31;
;  tage[11]:= 30; tage[12]:= 31;
;  device:= "1";
;  poke 53280,0; poke 53281,0;
;  writeln ("[illegible]* Eingabe des Tagesdatums in der");
;  writeln ("[illegible]* Form tt.mm.jjjj");
;  for lauf:= 1 to 15000 do begin end;
;  tagdatum:= "tt.mm.jjjj";
;  dateneingabe (t,m,j,c);
;  tagdatum:= datum;
;  for lauf:= 1 to max do initial (lauf);
;  dateiname:= "name eingeben!";
;  bottomstring:= "└─────────────────────────────────────┘";
;
repeat
if device="8" then reset (error);
write ("[illegible]",chr(142),"[illegible] UCCCCCCCC");
write ("CCCCCCCCCCCCCCCCCCCCCCCCCC");
write ("CCCI  I[illegible]++ terminkalender");
write (" ++ hauptmenue +++[illegible]I  JCCC");
```

```
;                 else termin.b[lauf]:= 0;
;               . eingabe;
;                 end;
;              3: begin
;                 tageingabe;
;                 termin.a[lauf]:= 1;
;                 termin.b[lauf]:= tagnummer;
;                 eingabe;
;                 end;
;           end; (* of case *)
;      end; (* of if *)
;      else begin
;         writeln ("[illegible] kein platz mehr frei!");
;         i2:= 2; for i1:= 1 to 10000 do begin end;
;      end; (* of else *)
;      until schluss(i2)="j";
"3":   repeat i2:= 1;
;      datenart;
;      case eingabeart of
;              0: i2:= 2;
;            1,2: begin dateneingabe (st,sm,sj,sc);
;                 for lauf:= 1 to max do
;                 if termin.a[lauf]=sc then begin
;                 if (eingabeart=1) and (termin.b[lauf]=sj) then
;                 datloeschen;
;                 if (eingabeart=2) and (termin.b[lauf]=0) then
;                 datloeschen;
;                 end; (* of if *)
;                 end;
;              3: begin
;                 tageingabe;
;                 for lauf:= 1 to max do
;                 if tagnummer=termin.b[lauf] then datloeschen;
;                 end;
;           end; (* of case *)
;      until schluss(i2)="j";
"4":   begin writeln ("[illegible]");
;      dateneingabe (st,sm,sj,sc);
;      reset (drucker);
;      writeln ("[illegible]    Mit der"," Stoptaste ins"," Hauptmenue    ");
;      repeat
;         wochentag (st,sm,sj,tagnummer);
;         gefunden[0]:= false;
;         for lauf:= 1 to max do begin
;           if test(lauf) then gefunden[0]:= true;
;           gefunden[lauf]:= test(lauf);
;         end; (* of for *)
;         if gefunden[0] then begin
;           write (drucker,"┌──",tagname[tagnummer]);
;           for i1:= length(tagname[tagnummer])+5 to 17 do
;           write (drucker,"─");
;           write (drucker,st);
;           if st<10 then write (drucker," ");
;           write (drucker,".",sm);
;           if sm<10 then write (drucker," ");
;           writeln (drucker,".",sj,"─────┐");
;        writeln (drucker,"|                                          |");
;           for lauf:= 1 to max do begin
;             if gefunden[lauf] then begin
;               formatausgabe;
;               writeln (drucker,string);
;             end; (* of if *)
;           end; (* of for *)
;           writeln (drucker,bottomstring);
;           writeln (drucker);
;         end; (* of if *)
```

```
        weiter;
        get druckstop;
      until druckstop<>chr(0);
      close (drucker);
      end; (* of case4 *)
"5": begin
      if sicher(5)="j" then begin
      for lauf:= 1 to max do initial (lauf);
      case device of
       "1": begin lauf:= 1;
            reset (cassin,dateiname);
            while not(eof(1)) do begin
              readln (cassin,termin.a[lauf]);
              readln (cassin,termin.b[lauf]);
              i1:= 0; repeat i1:= i1+1;
              read (cassin,termin.c[lauf,i1]);
              until ((termin.c[lauf,i1]=chr(13)) or (i1=37));
       if termin.c[lauf,i1]=chr(13) then termin.c[lauf,i1]:= chr(0);
       else read (cassin,null);
              lauf:= lauf+1;
            end; (* while *)
            close (cassin);
            end; (* "1" *)
       "7": begin lauf:= 1;
            reset (turboin,dateiname);
            while not(eof(1)) do begin
              read (turboin,z1);
              read (turboin,z2);
              termin.a[lauf]:= ord(z1)+256*ord(z2);
              read (turboin,z1);
              read (turboin,z2);
              termin.b[lauf]:= ord(z1)+256*ord(z2);
              for i1:= 1 to 37 do
              read (turboin,termin.c[lauf,i1]);
              lauf:= lauf+1;
            end; (* while *)
            close (turboin);
            end; (* "7" *)
       "8": begin
            floppystring[1..0]:= dateiname;
            floppystring[15..18]:= ",s,r";
            floppystring[19]:= chr(0);
            writeln ("loading: ",floppystring[1..14]);
            reset (floppy,floppystring);
            if not(ioerror(1)) then begin
            lauf:= 1;
            while not(eof(1)) do begin
              read (floppy,termin[lauf]);
              lauf:= lauf+1;
            end; (* of while *)
            end; (* of if *)
            close (floppy);
            if ioerror(1) then begin end;
          end; (* "8" *)
        end; (* of case *)
      end; (* of if *)
      end; (* of case 5 *)
"6": begin
      write ("dateiname: -> ");
      stringinput (14);
      dateiname:= string[1..14];
      i1:= length(dateiname);
      if i1<14 then begin
         i1:= i1+1;
         for i2:= i1 to 14 do dateiname[i2]:= chr(160);
      end; (* of if *)
      end; (* of case 6 *)

"7": begin
      if sicher(7)="j" then begin
      for lauf:= 1 to max do
      begin
        if ((termin.b[lauf]<j) and (termin.b[lauf]>7)) then
        initial (lauf);
        else begin
          if ((termin.a[lauf]<c) and (termin.b[lauf]=j)) then
          initial (lauf);
        end; (* of else *)
      end; (* of for *)
      case device of
        "1": begin rewrite (cassout,dateiname);
             for lauf:= 1 to max do
             if termin.a[lauf]<>0 then begin
                writeln (cassout,termin.a[lauf]);
                writeln (cassout,termin.b[lauf]);
                writeln (cassout,termin.c[lauf]);
             end; (* of if *)
             close (cassout);
             end; (* "1" *)
        "7": begin rewrite (turboout,dateiname);
             for lauf:= 1 to max do begin
             if termin.a[lauf]<>0 then begin
                write (turboout,chr(termin.a[lauf] mod 256));
                write (turboout,chr(termin.a[lauf] div 256));
                write (turboout,chr(termin.b[lauf] mod 256));
                write (turboout,chr(termin.b[lauf] div 256));
                for i1:= 1 to 37 do
                write (turboout,termin.c[lauf,i1]);
             end; (* of if *)
             end; (* of for *)
             close (turboout);
             end; (* "7" *)
        "8": begin
             floppystring[1..0]:= "@:";
             floppystring[3..0]:= dateiname;
             floppystring[17..0]:= ",s,w";
             floppystring[21]:= chr(0);
             writeln ("saving: ",floppystring[1..16]);
             rewrite (floppy,floppystring);
             if not(ioerror(1)) then begin
             for lauf:= 1 to max do
               if termin.a[lauf]<>0 then
                  write (floppy,termin[lauf]);
             end; (* of if *)
             close (floppy);
             if ioerror(1) then begin end;
             end; (* "8" *)
             end; (* of case *);
             end; (* of if *)
             end; (* of case 7 *)
        "8": begin
             if (sicher(8)="j") then menuewahl:= "q";
             end;

      end; (* of maincase *)
  close (error);
until menuewahl="q";

real;
goto start;

end.
```

SUPERKÄSI

Dieses Spiel hat nichts mit irgendwelchen Käsesorten zu tun, auch wenn der Name so klingen mag. Vielmehr ist SUPERKÄSI ein Spiel, das Sie sicherlich noch aus Ihrer Schulzeit kennen (Käsekästchen). Ihr Gegner ist hierbei aber der Computer.

Als Erstes sehen Sie das Hauptmenü. Hier können Sie wählen, wieviel "Striche" der Computer vorgeben, mit welcher Spielstärke gespielt werden und wer anfangen soll.

Nachdem Sie die Optionen im Hauptmenü gesetzt und das Spiel gestartet haben, sehen Sie das Spielfeld. Es ist einem karierten Blatt Papier nachempfunden. Fängt der Computer mit dem Spiel an, setzt er eine schwarze Linie. Wird auf dem Spielfeld eine Hand abgebildet, setzen Sie eine Linie.

Beispiel:

Angenommen, der Computer setzt eine waagerechte Linie, Sie setzen an diese waagerechte Linie eine senkrechte Linie und der Computer setzt seine Linie so, daß von einem Quadrat nur noch eine Seite offen bleibt.

Als nächstes könnten Sie mit einer weiteren Linie dieses Quadrat vervollkommnen und Ihr Zeichen in das Quadrat eintragen. Sie hätten damit Ihren ersten Punkt zu verzeichnen.

Spielsteuerung:

Das Spiel kann mit dem Joystick in Port 2 oder der Tastatur gespielt werden. Um eine ausgewählte Linie zu markieren, müssen Sie zunächst die abgebildete Hand in ein daran angrenzendes Kästchen manövrieren. Dann drücken Sie den Feuerknopf, und die Hand wechselt Ihre Farbe. Jetzt können Sie eine der vier Linien, die das Kästchen Kästchen umgrenzen, durch erneutes Verschieben der Hand auswählen, und durch einen weiteren Tastendruck auf den Feuerknopf kennzeichnen.

Tastaturbelegung:

Unten = M
Oben = K
Rechts = C
Links = Z
Feuerknopf = E

Mathe mit Nico

Diesmal treffen Sie auf zwei Unbekannte. Sie müssen mit Ihnen rechnen, um sie zu entlarven.

Für die Lösung von Gleichungen mit zwei Unbekannten in dieser Folge wurde die Rechenseite erweitert. Wie in den früheren Folgen, steht Ihnen innerhalb des Kurses eine Bildschirmseite zur Verfügung. Auf dieser Seite können Sie Rechnungen durchführen, wobei der C 64 Ihnen als "Taschen-Rechner" zur Seite steht.

Ab dieser Ausgabe dürfen Sie jetzt alle Rechen-Kommandos verwenden, wie Sie sie aus dem Direkt-Modus des C 64 kennen. Sogar die Verwendung der Standard-Funktionen wie SIN, EXP, INT, und so weiter sind möglich.

Sie geben beispielsweise ein:

8*INT(4096/100) RETURN-Taste

und der Rechner gibt aus:

= 320

Und nun viel Spaß bei "Mathe mit Nico"!

REASSEMBLER V7.6

Was ist ein REASSEMBLER?

Das Problem ist aus dem Umgang mit Monitorprogrammen bekannt: Sie haben ein Maschinen-Programm im Speicher oder auf Diskette und möchten sich gerne den entsprechenden Assembler-Code ansehen, um zu erfahren, mit welchen Tricks und welcher Weisheit das betreffende Programm erstellt wurde, und an welcher Stelle es am sinnvollsten geändert werden kann. Doch liefern Monitore in der Regel nur eine direkte Umwandlung sinnvoller Codes in Assembler-Befehle, sogenannte MNEMONICS.

Die Übersicht wird zum einen erheblich dadurch erschwert, daß keinerlei Label aufgeführt werden und zum anderen dadurch, daß ein Monitor keine Tabellen erkennen kann, die zum Beispiel Texte enthalten, also keinen sinnvollen OP-Code.

Hierfür gibt es nun sogenannte Reassembler. Diese Programme erzeugen aus einem Maschinen-Code einen lesbaren Assembler-Code, zumindest sollten sie das. In den meisten Fällen können die erzeugten Assembler-Listings auch auf einen Drucker ausgegeben werden, doch fehlt oft die Möglichkeit, diese Quellen-Codes auf den eigenen Datenträger

abzuspeichern und anschließend von einem Assembler nach eventuellen Änderungen assemblieren zu lassen.

Ein weiteres Problem ergibt sich, wenn das Maschinen-Programm ausgerechnet den Speicherbereich beansprucht, in dem der Reassembler selbst liegt. In diesem Fall steht der neugierige Anwender im Dunkeln. Ein Reassembler sollte auch in der Lage sein, direkt von Diskette herunter zu reassemblieren und den Source-Code (Quellen-Liste) wieder auf Diskette auszugeben. Nur so können längere Sources verarbeiten werden, ohne den Speicherbereich zu überziehen. Hierbei muß das Programm soweit abgesichert sein, daß fehlerhafte Operationen der Disketten- oder Kassetten-Station nicht zum Absturz führen.

Funkionsweise des ReAss

ReAss bearbeitet ein Objekt-File (Maschinen-Programm) in zwei Durchgängen, diese werden PASSES genannt:

PASS 1:

Hier werden alle Adressen (Operanden), die auf das Programm selbst zeigen, als Label dargestellt.

Beginnt ein Maschinen-Programm bei $C000 und endet bei $C500, so wird aus:

JSR $C123 ein JSR LBXXX

wobei 'LB' die Standard-Bezeichnung für Label und 'XXX' eine fortlaufende Nummerierung darstellen. Adressen außerhalb des Programms werden nicht in Label umgewandelt. Wurde eine Symboltabelle zugeladen, und ist dieses Label in der Tabelle enthalten, wird 'LBXXX' gegen die aufgeführte Bezeichnung ausgetauscht.

Beispiel:

'LIESRT = $c123' wird zu JSR LIESRT

Sind alle Label in PASS 1 vom ReAss erkannt worden, werden sie in aufsteigender Reihenfolge sortiert, was das Auffinden im Source-Listing erheblich erleichtert.

PASS 2:

Im PASS 2 wird der eigentliche Source-Code erzeugt. Dadurch, daß in PASS 2 auch Tabellen erkannt werden und der Reassembler jede Adresse auf Label überprüfen muß, dauert Durchgang 2 natürlich länger als der vorherige!

Der ReAss erkennt Tabellen nur aus dem logischen Zusammenhang. Er simuliert das Programm nicht, wie das Relocator-Programme tun. (In der

nächsten Ausgabe werden wir so ein Programm veröffentlichen!!) Daraus folgt, daß indirekte Zeiger (zum Beispiel auf Texte) nicht erkannt werden und daß dort keine Label gesetzt werden können. Diese müßten dann vom Benutzer selbst ergänzt werden.

Steht im Maschinen-Programm:

```
C123   A9 4C 85 2C
```

so wird daraus, falls $C123 innerhalb des Programms angesprungen wird, beispielsweise:

```
1210↑ LB021  LDA #$4C
1220↑        STA $2C
```

Wurde $2C in einer Symblotabelle mit BASANH = $2C definiert, erscheint:

```
1210↑ LB021  LDA #$4C
1220↑        STA BASANH
```

Sie können so durch geschickte Definitionen in einer Symboltabelle die Lesbarkeit Ihres Source-Codes noch erhöhen.

Das Zeichen '↑' wird als Formatzeichen am Anfang jeder Zeile verwendet und soll den BASIC-Editor daran hindern, eventuelle folgende Leerzeichen zu ignorieren.

Die Bedienung des ReAss

Unser Reassembler sollte nicht nur für einen speziellen Assembler geeignet sein. Deshalb müssen vom Benutzer einige Definitions-Zeilen, die sich auf den verwendeten Assembler beziehen, selbst eingegeben werden. Hierbei ist zu beachten, daß ReAss nur mit Assemblern verträglich ist, die den BASIC-Editor zur Bearbeitung der Source-Codes benutzen. Assembler mit eigenen Editoren oder eigenen Zeilenformaten können nicht direkt verwendet werden.

Folgende drei Zeilen beziehen sich auf diese Anpassung:

- Definitionszeile: legt den Aufruf des jeweiligen Assemblers fest (zum

Beispiel: SYS 49847)

- Verknüpfungzeile: sorgt dafür, daß bei längeren Source-Codes (Quellen), die nicht mehr in den RAM-Speicher des C 64 passen, eine Verknüpfung der einzelnen Files hergestellt werden kann (zum Beispiel durch .FILE...)

- Endzeile: Manche Assembler benötigen eine Abschluß-Zeile der Art .END.....

Beziehungen zur Peripherie

Für ReAss existieren zwei Möglichkeiten auf ein Maschinen-Programm zuzugreifen:

1. direkt aus dem Speicher

2. von einem Daten-Träger

Objekte im Speicher

Soll der Objekt-Code eines Maschinen-Programms direkt aus dem Speicher reassembliert werden, muß das betreffende Programm vorher im Speicher stehen. Dabei darf das Programm natürlich nicht in den Bereichen liegen, die ReAss selbst benutzt. Freie Bereiche sind:

$9100-$9fff, $c000-$d000 und $ea00-$fffa.

Der Aufruf:

SYS 32771, Anfang, Ende, Start, Gerät, AM

Anfang, Ende und Start sind Adressen. Anfang und Ende beziehen sich auf den Bereich, in dem das Maschinen-Programm im Speicher liegt.

Start ist der Beginn des Speicherbereichs, in den das Programm später assembliert werden soll. Sprünge und Aufrufe im Maschinen-Programm, die innerhalb des Bereichs zwischen Start und dem neuen Ende liegen (Start + Programmlänge), werden mit Labeln versehen. Soll das Programm die alte Start-Adresse behalten ist Anfangs- gleich Start-Adresse.

Beispiel:

sys32771,38000,39000,40960,8,55

Das Programm im Bereich von 38000 bis 39000 wird nach 40969 (bis 41960) reassembliert. Ein JMP $A100 (JMP 40216) würde in einen JMP LBXXX umgewandelt.

Gerät bezieht sich auf den gewünschten Datenträger für den Source-Code. Es stehen zur Verfügung:

1 - Kassette im Commodore-Format

7 - Kassette im SuperTape-Format

8 - Diskette im DOS-Format.

Wird 7 gewählt, muß vorher SuperTape gestartet werden. Dadurch geht der Bereich $C000-$C750 jedoch verloren.

AM ist ein Wert für Spezis, denn er verändert die Adresse 01 in der Zero-Page, was zu erstaunlichen Effekten führt, da hiermit unter anderem das Betriebssystem stillgelegt werden kann. Der Standard-Wert ist '55'.

ReAss verkraftet aber andere Werte, da intern dieser Wert nur zur Umschaltung zwischen ROM und RAM benutzt wird. Wer ein Programm aus dem RAM unter dem Kernal lesen möchte, muß hier den entsprechenden Wert eintragen. Zum Beispiel: '53', wodurch Bit 1 auf '0' gesetzt wird und der RAM-Bereich unter dem KERNAL- und BASIC-ROM angesprochen werden kann.

Grundsätzlich ist zu beachten, daß der Speicherbereich für den Source-Code nicht unbegrenzt ist. Ist der Source-Code zu lang (größer 30719 Byte), wird in jedem Fall auf den Datenträger ausgelagert.

Im Kassetten-Betrieb kann jedoch nur ein Source-File abgespeichert werden. Dies ist sinnvoll, da die meisten Assembler bei Kassetten-Operationen keine Verknüpfung durchführen können.

Achten Sie im Kassetten-Betrieb darauf, daß Ihr Source-Code nicht zu lang wird! Sollte ein Speicher-Überlauf entstehen, versuchen Sie, die Source zu verkürzen, indem Sie zum Beispiel die Trennzeilen verkürzen oder mehrere Befehle in einer Zeile zulassen.

Träger-Objekte

Längere Objekt-Files können nur von Diskette gelesen werden. Die gleiche Diskette dient dann auch zum Abspeichern des reassemblierten Source-Codes. Also: Denken Sie daran, daß auf der Diskette genügend freie Blöcke vorhanden sein müssen.

Zum Aufruf geben Sie ein:

SYS 32768

Im Bilde des ReAss

Haben Sie den ReAss über den entsprechenden SYS-Befehl aufgerufen, erscheint eine Bildschirm-Maske. In der oberen Hälfte werden die wichtigsten Festlegungen, die Sie getroffen haben, protokolliert.
In der 22. Zeile erscheinen die Aufforderungen des ReAss-Systems zur Bedienung Ihres Peripherie-Geräts.

In der 23. Zeile wird eine Status-Meldung im Disketten-Betrieb angezeigt.

In der 24. Zeile wird mitgeteilt, welche Eingabe erforderlich ist.

Die letzte Zeile dient der Eingabe.

Der ReAss stellt Sie vor folgende Entscheidungen:

Auf Gerät Nr. J/N (obere Zeile)

Sie entscheiden, ob der Source-Code in jedem Fall auf den entsprechenden Datenträger geschrieben werden soll.

Schreibfile (in Zeile 24)

Hierunter legen Sie den Namen fest, unter dem das Source-File auf dem Datenträger gespeichert werden soll. Diese Festlegung ist auf jeden Fall erforderlich, auch wenn Sie vorher mit 'N' geantwortet haben. Falls der Source-Code zu lang werden sollte, werden die Source-Files unter dem Namen abgespeichert, wobei dem Namen eine fortlaufende Kennziffer vorangestellt wird.

Trennen J/N (obere Zeilen)

Um Ihr Source-Listing lesbarer zu gestalten, können Sie nach 'J' in der letzten Zeile eine Zeichen-Kette eingeben, die im Listing als Trennzeile verwendet wird. Trennzeilen werden nach den Befehlen

BRK, JMP, RTS und RTI

eingesetzt.

Haben Sie 'N' gewählt, wird keine Eingabe erwartet.

Kopfzeile (Zeile 24)

In der Kopfzeile können Sie das Source-File Ihrem Assembler anpassen, indem Sie die notwendigen Start-Befehle festlegen.

Beispiel:

←9*4096:.opt p:*= £

Diese Zeile enthält zwei Sonderzeichen, die zur Steuerung des Editors und zur automatischen Zuweisung dienen:

← ersetzt den BASIC-Befehl SYS.

£ gilt als Joker für die Start-Adresse, die entweder im SYS-Befehl angegeben wurde oder bei SYS32768 von Diskette gelesen wird. Diese Zeile erscheint im Listing beispielsweise als:

sys39864:.opt p:*=$c000

Verk.-Zeile (Zeile 24)

Die Verknüpfungs-Zeile dient dazu, die Verbindung zwischen mehreren

Source-Files für Ihren Assembler herzustellen. Sie könnte lauten:

.file 8,"£"

wobei '£' wiederum ein Joker ist, für den im Listing der Name des Schreibfiles mit führender Kennziffer eingesetzt wird. Die Kennziffer dient zum Durchnummerieren meherer Files.

Endzeile (Zeile 24)

Einige Assembler erwarten als Abschluß des gesamten Source eine spezielle Endzeile. Zum Beispiel:

.end 8,"£"

Für den Joker gilt das Gleiche wie oben. Eine Verknüpfung über mehrere Files ist für den Kassetten-Betrieb nicht sinnvoll (Sie müßten nämlich hin-her-spulen) und daher nicht vorgesehen.

1 Bef. pro Zeile J/N (obere Zeile)

Sie können mit 'J' ein Listing erzeugen, in dem nur ein Befehl je Zeile aufgeführt ist. Bei 'N' werden die Befehle mit ':' verknüpft. Es werden jedoch nur Zeilen von 40 Zeichen Länge erzeugt.

Beachten Sie, daß bei '1 Befehl pro Zeile' das Source-File erheblich länger wird!

Symbtab laden J/N (obere Zeile)

Für manche Assembler können eigene Symboltabellen erzeugt werden. In einer solchen Tabelle können Sie Label und Variablen mit Namen belegen. Beim Reassemblieren werden dann die entsprechenden Adressen durch die jeweiligen Namen ersetzt. Dadurch kann die Lesbarkeit eines Assembler-Programms erheblich verbessert werden.

Im Anschluß finden Sie ein Beispiel für eine Symboltabelle, die sich an den Namensgebungen in den Commodore-Veröffentlichungen orientiert.

Beachten Sie:

Symboltabellen müssen assemblierte Tabellen Ihres Assemblers sein! Sie können hier keine Source-Symboltabelle zuladen. Außerdem darf eine Symboltabelle nicht zu groß sein (kleiner als 4 Blöcke, entspricht 1023 Byte).

Objekte auf Diskette

Erwartet der ReAss das Objekt-File von Diskette, erfolgt noch die Frage:

Lesefile (Zeile 24)

Hier geben Sie den Namen ihres Objekt-Files auf Ihrer Diskette an, das Sie reassemblieren wollen.

Zusatz-Service

ReAss stellt Ihnen noch einige zusätzliche Möglichkeiten zur Verfügung:

Belegung der Funktionstasten:

F1: Senden von Kommandos an die Floppy.

F3: Ausgabe der Direktory.

F5: Anzeige des DOS-Status

F7: Abbruch Direktory-Anzeige im Disk-Modus (sonst siehe Editor-Funktion)

Die Disk-Kommandos entsprechen der String-Syntax innerhalb der PRINT# oder OPENnn,8,mm, Befehle.

Beispiel:

Betätigen Sie F1, erscheint DISK (Zeile 24)

Geben Sie jetzt beispielsweise 'UJ' ein, bewirkt dies ein Reset des DOS Ihrer Floppy. Weitere Möglichkeiten entnehmen Sie bitte Ihrem Floppy-Handbuch.

Nach der obengenannten Sequenz erscheint dort nach Druck auf F5 die Einschalt-Meldung des DOS-Betriebssystems.

Die Direktory wird in der zweiten unteren Hälfte des Bildschirm-Fensters angezeigt. Sie kann mit der Leertaste weitergeblättert oder mit F7 abgebrochen werden.

Zeilen-Editor

Während der Eingabe in der Zeile 25 können Sie mit den Cursor-Tasten für Rechts und Links Ihren eingegebenen Text korrigieren.

CLR/HOME löscht die Eingabe vollständig

INST/DEL löscht ab Cursor-Position nach rechts.

F7 schaltet den Hochkomma-Modus wieder aus.

Fortsetzung auf Seite 32

Symboltabelle Reassembler

```
1000 open4,4,7
1010 sys32768
1020 *= $6000
1030 .opt p4,oo
1031 ; s y m b o l t a b e l l e
1032 ; ========================
1033 ;
1034 ; diese  symboltabelle kann nach
1035 ; bedarf in teilen zum assemblie-
1036 ; ren mit 'profi-ass' zugeladen
1037 ; werden. die gesamte tabelle ist
1038 ; zu gross!
1039 ; die bezeichnungen orienteren
1040 ; sich an veroeffentlichungen des
1041 ; hauses commodore.
1042 ;
1050 ;"——————————page 0——————————————
1051 ;
1052 ; die beruehmte zero-page. siehe
1053 ; handbuch zum c 64.
1054 ;
1060 ;
1070 d6510  = $00  :r6510  = $01
1080 adray1 = $03  :adray2 = $05
1090 charac = $07  :endchr = $08
1100 trmpos = $09  :verck  = $0a
1110 count  = $0b  :dimflg = $0c
1120 valtyp = $0d  :intflg = $0e
1130 garbfl = $0f  :subflg = $10
1140 inpflg = $11  :tansgn = $12
1150 linnum = $14  :temppt = $16
1160 lastpt = $17  :tempst = $19
1170 index  = $22  :resho  = $26
1180 txttab = $2b  :vartab = $2d
1190 arytab = $2f  :strend = $31
1200 fretop = $33  :frespc = $35
1210 memsiz = $37  :curlin = $39
1220 oldlin = $3b  :oldtxt = $3d
1230 datlin = $3f  :datptr = $41
1240 inpptr = $43  :varnam = $45
1250 varpnt = $47  :forpnt = $49
1260 tempnt = $4b  :facexp = $61
1270 facho  = $62  :facsgn = $66
```

```
1683 ; be oder prozessorstack.
1690 ;
1700 bad    = $0100:pstk   = $0100
1710 ;
1720 ;"————————— page 2 ———————————————
1721 ;
1722 ; diese page wird ebenfalls wie
1723 ; zeropage von einigen kernalrou-
1724 ; tinen benutzt.
1730 ;
1740 buf    = $0200:lat    = $0259
1750 fat    = $0263:keyd   = $0277
1760 memstr = $0281:memsiz = $0283
1770 timout = $0285:color  = $0286
1780 gdcol  = $0287:hibase = $0288
1790 xmax   = $0289:rptflg = $028a
1800 kount  = $028b:delay  = $028c
1810 shflag = $028d:lstshf = $028e
1820 keylog = $028f:mode   = $0291
1830 autodn = $0292:m51ctr = $0293
1840 m51cdr = $0294:m51ajb = $0295
1850 rsstat = $0297:bitnum = $0298
1860 baudof = $0299:ridbs  = $029c
1870 rodbs  = $029d:rodbe  = $029e
1880 irqtmp = $029f:enabl  = $02a1
1890 ;
1900 ;"——————————page 3————————————————
1910 ;
1911 ; diese page enthaelt vektoren
1912 ; fuer einspruenge in kernal- und
1913 ; basic-routinen. im bereich $033c
1914 ; bis $03fb liegt der bandpuffer.
1915 ;
1920 ierror = $0300:imain  = $0302
1930 icrnch = $0304:iqplop = $0306
1940 igone  = $0308:ieval  = $030a
1950 sareg  = $030c:sxreg  = $030d
1960 syreg  = $030e:spreg  = $030f
1970 usrpok = $0310:usradd = $0311
1980 cinv   = $0314:cbinv  = $0316
1990 nminv  = $0318:iopen  = $031a
2000 iclose = $031c:ichkin = $031e
2010 ickout = $0320:iclrch = $0322
```

```
2330 spmul  = $d01c:spexx  = $d01d
2340 spcsp  = $d01e:spcbk  = $d01f
2350 bocol  = $d020:bcol1  = $d021
2360 bcol2  = $d022:bcol3  = $d023
2370 bcol4  = $d024:spmul0 = $d025
2380 spmul1 = $d026:sp0col = $d027
2390 sp1col = $d028:sp2col = $d029
2400 sp3col = $d02a:sp4col = $d02b
2410 sp5col = $d02c:sp6col = $d02d
2420 sp7col = $d02e
2430 ;--------------------------------
2440 v1frql = $d400:v1frqh = $d401
2450 v1pwl  = $d402:v1pwh  = $d403
2460 v1crg  = $d404
2470 env1ad = $d405:env1sr = $d406
2480 v2frql = $d407:v2frqh = $d408
2490 v2pwl  = $d409:v2pwh  = $d40a
2500 v2crg  = $d40b
2510 en2vad = $d40c:en2vsr = $d40d
2520 v3frql = $d40e:v3frqh = $d40f
2530 v3pwl  = $d410:v3pwh  = $d411
2540 v3crg  = $d412
2550 en3vad = $d413:en3vsr = $d414
2560 ficfl  = $d415:ficfh  = $d416
2570 firvc  = $d417:fimov  = $d418
2580 adcv1  = $d419:adcv2  = $d41a
2590 osrdm  = $d41b:en3ou  = $d41c
2600 ;--------------------------------
2610 dat1pa = $dc00:dat1pb = $dc01
2620 d1adir = $dc02:d1bdir = $dc03
2630 tim1al = $dc04:tim1ah = $dc05
2640 tim1bl = $dc06:tim1bh = $dc07
2650 tod1ds = $dc08:tod1s  = $dc09
2660 tod1m  = $dc0a:tod1h  = $dc0b
2670 ssio1  = $dc0c:ciaiq1 = $dc0d
2680 cia1ca = $dc0e:cia1cb = $dc0f
2690 ;--------------------------------
2700 dat2pa = $dd00:dat2pb = $dd01
2710 d2adir = $dd02:d2bdir = $dd03
2720 tim2al = $dd04:tim2ah = $dd05
2730 tim2bl = $dd06:tim2bh = $dd07
2740 tod2ds = $dd08:tod2s  = $dd09
2750 tod2m  = $dd0a:tod2h  = $dd0b
```

```
1280 sgnflg - $67   :bits   - $68
1290 argexp - $69   :archo  - $6a
1300 argsgn - $6e   :arisgn - $6f
1310 facov  - $70   :fbufpt - $71
1320 chrget - $73   :chrgot - $79
1330 txtptr - $7a   :rndx   - $8b
1340 status - $90   :stkey  - $91
1350 svxt   - $92   :verck  - $93
1360 c3po   - $94   :bsour  - $95
1370 syno   - $96   :ldtnd  - $98
1380 dfltn  - $99   :dflto  - $9a
1390 prty   - $9b   :dpsw   - $9c
1400 msgflg - $9d   :ptr1   - $9e
1410 ptr2   - $9f   :time   - $a0
1420 cntdn  - $a5   :bufpnt - $a6
1430 inbit  - $a7   :bitci  - $a8
1440 rinone - $a9   :ridata - $aa
1450 riprty - $ab   :sal    - $ac
1460 eal    - $ae   :cmp0   - $b0
1470 tape1  - $b2   :bitts  - $b4
1480 nxtbit - $b5   :rodata - $b6
1490 fnlen  - $b7   :la     - $b8
1500 sa     - $b9   :fa     - $ba
1510 fnadr  - $bb   :roprty - $bd
1520 fsblk  - $be   :mych   - $bf
1530 cas1   - $c0   :stal   - $c1
1540 memuss - $c3   :lstx   - $c5
1550 ndx    - $c6   :rvs    - $c7
1560 indx   - $c8   :lxsp   - $c9
1570 sfdx   - $cb   :blnsw  - $cc
1580 blnct  - $cd   :gdbln  - $ce
1590 blnon  - $cf   :crsw   - $d0
1600 pnt    - $d1   :pntr   - $d3
1610 qtsw   - $d4   :lnmx   - $d5
1620 tblx   - $d6   :insrt  - $d8
1630 ldtb1  - $d9   :user   - $f3
1640 keytab - $f5   :ribuf  - $f7
1650 robuf  - $f9   :frekzp - $fb
1660 baszpt - $ff
1670 ;
1680 ;"———————— page 1 ————————————
1681 ;
1682 ; entweder speicher bei bandeinga-

2020 ibasin - $0324:ibsout - $0326
2030 istop  - $0328:igetin - $032a
2040 iclall - $032c:usrcmd - $032e
2050 iload  - $0330:isave  - $0332
2060 tbuffr - $033c
2070 ;
2080 ;"——————————page 4-7——————————
2090 ;
2091 ; der bildschirmbereich und die
2092 ; spritepointer
2093 ;
2100 vicscn - $0400
2101 spptn1 - $07f8:spptn2 - $07f9
2102 spptn3 - $07fa:spptn4 - $07fb
2103 spptn5 - $07fc:spptn6 - $07fd
2104 spptn7 - $07fe:spptn8 - $07ff
2110 ;
2120 ;"——————————page 208——————————
2130 ;
2131 ; die adressen des vic ab $d000
2132 ;                 des sid ab $d400
2133 ; und der beiden cias:
2134 ; cia1 $dc00 und cia2  $dd00
2135 ;
2140 vic    - $d000:v      - vic
2150 sid    - $d400:s      - sid
2160 cia1   - $dc00:c1     - cia1
2170 cia2   - $dd00:c2     - cia2
2180 ;---------------------------------
2190 s0px   - $d000:s0py   - $d001
2200 s1px   - $d002:s1py   - $d002
2210 s2px   - $d004:s2py   - $d005
2220 s3px   - $d006:s3py   - $d007
2230 s4px   - $d008:s4py   - $d009
2240 s5px   - $d00a:s5py   - $d00b
2250 s6px   - $d00c:s6py   - $d00d
2260 s7px   - $d00e:s7py   - $d00f
2270 smsbx  - $d010:vicr1  - $d011
2280 rairq  - $d012:ligx   - $d013
2290 ligy   - $d014:spen   - $d015
2300 vicr2  - $d016:spexy  - $d017
2310 vimcr  - $d018:viirq  - $d019
2320 irqms  - $d01a:spbck  - $d01b

2760 ssio2  - $dd0c:ciaiq2 - $dd0d
2770 cia2ca - $dd0e:cia2cb - $dd0f
2780 ;
2790 ;"——————————page 255——————————
2800 ;
2810 cint   - $ff81:ioinit - $ff84
2820 ramtas - $ff87:restor - $ff8a
2830 vector - $ff8d:setmsg - $ff90
2840 second - $ff93:tksa   - $ff96
2850 memtop - $ff99:membot - $ff9c
2860 scnkey - $ff9f:settmo - $ffa2
2870 acptr  - $ffa5:ciout  - $ffa8
2880 untlk  - $ffab:unlsn  - $ffae
2890 listen - $ffb1:talk   - $ffb4
2900 readst - $ffb7:setlfs - $ffba
2910 setnam - $ffbd:open   - $ffc0
2920 close  - $ffc3:chkin  - $ffc6
2930 chkout - $ffc9:clrchn - $ffcc
2940 chrin  - $ffcf:chrout - $ffd2
2950 load   - $ffd5:save   - $ffd8
2960 settim - $ffdb:rdtim  - $ffde
2970 stop   - $ffe1:getin  - $ffe4
2980 clall  - $ffe7:tksa   - $ffea
2990 screen - $ffed:plot   - $fff0
3000 iobase - $fff3:
3010 nmi    - $fffa:reset  - $fffc
3020 irq    - $fffe
3030 ;
3031 ;!!!! achtung !!!!!!
3032 ; vergessen sie nicht, die ueber-
3033 ; fluessigen definitionen fuer
3034 ; ihre anwendung zu loeschen.
3035 ; sonst erfolgt ein:
3036 ; symboltab. overflow
3037 ;
3040 ;"——————————————————————————
3050 ;.sst 8,1,"kernal.sym"
3060 ;"——————————————————————————

ready.
```

Was bringen 80-Zeichen-Karten?

Für Textverarbeitung oder eine umfangreiche Tabellenkalkulation ist der Bildschirm des C-64 mit seinen 40 Zeichen pro Zeile oft zu klein. Um die endgültige Gestaltung eines Schriftstückes beurteilen zu können, wäre eine Schreibbreite, die der des Druckers entspricht, optimal. Für den Commodore 64 werden verschiedene 80-Zeichen-Karten angeboten. Können Sie hier Abhilfe schaffen? Die Problematik soll am Beispiel der von der Firma ROOS gelieferten Karte geschildert werden.

Man erhält eine Steckplatine, die in den Erweiterungsport des C-64 gesteckt wird und von der nunmehr das dem Monitor zugeführte Video-Signal abgegriffen wird. Dazu enthält diese Karte mehrere Ausgänge, eine Cinch-Buchse für das 80-Zeichen Composite-Video-Signal und zwei DIN-Buchsen, eine davon Ausgang für einen Chrominanz/Luminanz-Farbmonitor. Die dritte Buchse ist ein Eingang; ihr kann das bisherige 40-Zeichen-Bild zugeführt werden. Doch dazu unten mehr. Die Platine wird 'nackt', das heißt ohne Gehäuse, ausgeliefert; viele der eingelöteten IC's (16 an der Zahl) sind säuberlich abgeschliffen: da ist man im Störungsfalle bedingungslos auf den Hersteller angewiesen. Mechanischen Halt findet die Platine auf zwei Schrauben, die mit einer Isolierhülle überzogen sind, damit wenigstens der Arbeitstisch nicht verkratzt.

Softwareseitig enthält die Karte einige Eigenschaften, die die Möglichkeiten des C-64 nicht unbeträchtlich erweitern:

Da ist zunächst die angesprochene Mischmöglichkeit des C-64 Standardbildschirms mit dem 80-Zeichen-Bildschirm, wodurch es nun gelingt, Texte und Grafikdarstellungen einfachst zu mischen und miteinander zu kombinieren.

Weiterhin kann man sich den vertikalen Zeilenabstand von 1 bis zu 8 Pixel einstellen und damit die Lesbarkeit mancher Texte und Listings bedeutend erhöhen. Weiter gibt es eine Feststellmöglichkeit für die obere (erste) Bildschirmzeile, in der man zum Beispiel vermerken kann, woran gerade gearbeitet wird und - last not least - ist die Einblendung einer Uhrzeit möglich. Sie wird nicht aus dem ungenauen TI$ gewonnen, sondern aus den CIA-Realtime-Registern, die mit Netzfrequenz getriggert werden und genauer als TI oder TI$ sind.

Da das bisherige Bildschirm-RAM nicht mehr benötigt wird (denn die 80-Zeichen-Karte verfügt über ein eigenes RAM), kann man den freien BASIC-Speicher durch POKE 642,4:SYS 58260 auf 39935 freie Bytes erweitern.

Eine eigene Stromversorgung ist für die 80-Zeichen-Karte nicht erforderlich. Nach allen Prüfungen hat sich die Karte als voll kompatibel zum Commodore-Zeichensatz und zum Commodore-Bildschirmeditor gezeigt.

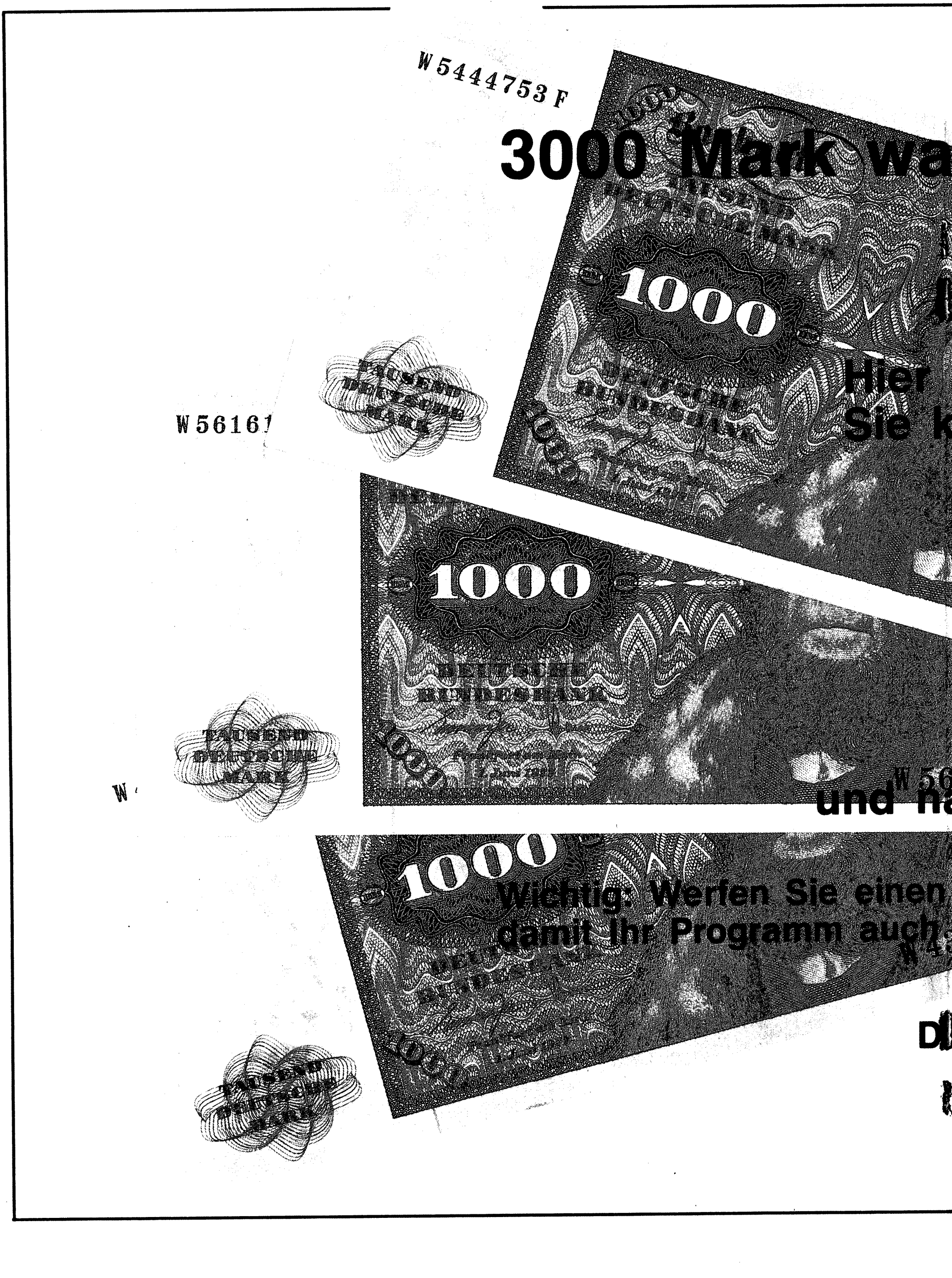
Daß der Karte eine Kassette bzw. Diskette mit einem deutschen Textverarbeitungsprogramm beiliegt, ist sowohl notwendig als auch erfreulich, denn es ist nicht möglich, eine der gängigen Tabellenkalkulationen (Multiplan, Calc Result...), Textverarbeitungen (Easy Script, Wordcraft, Wordpro, Vizawrite..) oder sonstigen Pakete mit dieser Karte zu betreiben, da sich alle Programme auf das 40-spaltige Format sowie die Originallage des Bildschirmspeichers beziehen. Das allerdings trübt die Freude an einem sonst so nützlichen Zubehörteil gewaltig, denn zumindest von meiner eingefahrenen und bestens bewährten Textverarbeitung möchte ich nicht herunter. ROOS umgeht das Problem elegant: im Gegensatz zu anderen Anbietern, bei denen man auf eventuelle eigene Phantasie angewiesen ist, werden zwei Listings für Konversionsprogramme von Vizawrite- und Easyscript-Files auf den ROOS-BASIC-Textverarbeiter mitgeliefert.

Bleiben noch einige Anmerkungen zur Zeichendarstellung. Der Original-C-64-Zeichensatz ist mit doppelten Pixelreihen aufgebaut und reichlich fett, da nur dies ein auf dem gewöhnlichen Fernseher lesbares Bild ergibt. Wer seinen Rechner an einem Monitor betreibt, wird sicher bald auf einen "schmalen" Zeichensatz umsteigen wollen, dabei allerdings feststellen, daß gleichzeitig die Anforderungen an den Monitor erheblich steigen. Dies ist bei den logischerweise noch schmaleren 80-Spalten-Zeichen noch ausgeprägter. Die ROOS-Karte liefert ein durchaus brauchbares und stabiles Bild. Ohne einen exzellenten Monitor ist der Betrieb einer solchen Karte allerdings von vorneherein ein sinnloses Unterfangen. Den Preis für die Anschaffung eines solchen Monitors müssen Sie also gleich mit einkalkulieren.

ES

Bezug:

DM 286,50
ROOS elektronik,
Kleiner Markt 7
4190 Kleve


W5444753F
3000 Mark wa
1000
TAUSEND DEUTSCHE MARK
Hier
Sie k
W56161
1000
DEUTSCHE BUNDESBANK
und na
1000
Wichtig: Werfen Sie einen
damit Ihr Programm auch



rten auf den Gewinner!

Der Wettbewerb geht weiter.

noch einmal kurz die Bedingungen:
önnen einsenden:

- Grafikprogramme
- Musikprogramme
- Spiele-
- Lernprogramme
- Anwenderprogramme

atürlich völlig neue Programmideen.

Blick in das Kapitel "Hinweise für Autoren",
nnerhalb von INPUT 64 lauffähig ist.

er Rechtsweg ist wie immer ausgeschlossen.

Fortsetzung von Seite 6

1. ABFRAGE: Nach der Eingabe des Datums errechnet das Programm den Wochentag und gibt eventuell vorhandene Termine aus. Mit einem Tastendruck können Sie weiterblättern, falls eine Bildschirmseite nicht ausreicht. Antworten Sie mit 'N' auf die Abfrage 'Schluß', wird der nächste Tag ausgegeben.

2. EINGABE: Sie können zuerst zwischen verschiedenen Terminarten wählen:

1. NORMALE TERMINE
2. DAUERTERMINE/JAHR
3. DAUERTERMINE/WOCHE

Nach der Festlegung geben Sie das Datum (1 und 2) oder den Wochentag ein (3). Anschließend geben Sie Ihren Termin-Text ein (max 37 Zeichen). Antworten Sie mit 'N' auf die 'Schluß-'Abfrage, bleiben Sie im Eingabe-Modus, bei 'J' kehrt das Programm ins Menue zurück

3. LOESCHEN: Sie beginnen wie oben unter 2. EINGABE. Wenn Sie auf die Abfrage 'Löschen' mit 'J' reagieren, wird der angezeigte Termin gelöscht und sein Speicherplatz freigegeben.

4. DRUCKEN: Sie können Ihren Terminkalender auf Device 4 ausgeben. Ist der Drucker nicht angeschlossen, erfolgt die Ausgabe auf den Bildschirm im gleichen Format. Die STOP-Taste bricht den Druckvorgang ab.

5. LADEN: Lädt eine Termin-Datei mit Namen (siehe 6) von Ihrem Datenträger. Der Speicher wird dabei überschrieben.

6. DATEINAME: Sie dürfen einen Namen mit maximal 14 Zeichen festlegen.

7. SAVEN: Nach der Sicherheitsabfrage werden die Termine aus dem Speicher auf dem Datenträger unter dem Namen (siehe 6) gesichert. Gleichnamige Dateien werden überschrieben.

8. NEUSTART: Sie könne mit einem neuen Tagesdatum beginnen.

@ DEVICE: Beim Start steht hier '1' (Kassette C=-Format). Sie können sich aber auch für 8 (Diskette) und 7 (Kassette SuperTape-Format) entscheiden. Bei 7 muß SuperTapige bringen.

64'er Tips

Wie angekündigt: Diesmal geht's rund um die INT-Funktion. Dabei können Sie sogar mit dem Zufall rechnen oder Pfennig-Fuchser um "halbe" Pfennige bringen.

Bitte abtrennen und im Briefumschlag versenden!

Kurs Komplett-Bestellschein

Ich bestelle den gesammelten Kurs BITS & BYTES IM VIDEO-CHIP

○ auf Kassette zum Preis von 17,80 DM

○ auf Diskette zum Preis von 24,80 DM

(Bitte ankreuzen/Nichtzutreffendes streichen.)

Absender und Lieferanschrift
Bitte in jedes Feld nur einen Druckbuchstaben (ä = ae, ö = oe, ü = ue)

Vorname/Zuname

Beruf/Funktion

Straße/Nr.

PLZ Wohnort

Datum/Unterschrift

Unterschrift ____________________
(Ein Verrechnungsscheck/quittierter Einzahlungsbeleg liegt bei)

Cave Climber

Bei diesem Spiel befinden Sie sich in einer Höhle. Nun müssen Sie versuchen, sich zum Ausgang am oberen Bildschirmrand durchzuhangeln. Dabei dürfen Sie nicht vergessen, die goldenen Bälle aufzusammeln.

Nur ist das nicht ganz einfach, denn Sie müssen beim Aufsammeln der Bälle den Höhlenbewohnern, Vögeln, Quarks und abgeworfenen Eiern ausweichen.

Gesteuert wird das Spiel mit dem Joystick in Port 2 oder der Tastatur.

Tastaturbelegung:

;	=	Rechts
:	=	Links
H	=	Oben
N	=	Unten
Space	=	Feuerknopf

Wenn Sie den Joystick nach oben drücken, wird, falls Sie festen Boden treffen, ein Seil gespannt (nur in Laufrichtung). Gehen Sie unter das Seil und drücken den Joystick nochmals nach oben, so können Sie sich am Seil entlanghangeln. Drücken Sie den Joystick nach unten, wird das Seil wieder losgelassen. Die Lianen, die von den einzelnen Plateaus herunterhängen, können Sie nur von oben erreichen.

Sie haben auch nur ein Seil zur Verfügung, deshalb wird das alte wieder eingezogen.

Möchten Sie gern eine Kaffeepause machen, können Sie das Spiel mit der Funktionstaste F7 unterbrechen. Durch Drücken des Feuerknopfes oder der Space-Taste, wird die Pause wieder aufgehoben.

Bemerkung:

- es gibt vier verschiedene Bilder, wobei das Spielen bis zum Level 25 immer mehr erschwert wird.

- nach jeder zweiten überstandenen Runde erhalten Sie ein "Extra-Leben".

- mit jedem aufgesammelten goldenen Ball wächst Ihre Energie, die Sie gut gebrauchen können.

MANAGER

In "meinem" Verein möchte ich einmal das Sagen haben!! Zuerst würde ich den Trainer rausschmeißen, dann würde ich andere und bessere Spieler verpflichten und überhaupt diese Eintrittspreise...

Die Spielidee

Jetzt können Sie - wenigstens auf der Simulationsebene - Ihren Lieblingsverein für die Saison 85/86 leiten. Die Rahmenbedingungen (alle Spielpaarungen, das Fassungsvermögen der Stadien und ein Anfangskapital) sind im Programm enthalten.

Sie können aber für "Ihren" Verein die folgenden Faktoren beeinflussen:

1. Spielstärke

Durch Kauf oder Verkauf von Spielern steigt oder sinkt die Spielstärke. (Beachten Sie aber beim Einkauf auch die Bilanz!)

2. Eintrittspreise

Der von Ihnen festgelegte Eintrittspreis (vom Programm wird 10,-DM vorgegeben), schlägt sich nicht nur - in Verbindung mit den Zuschauerzahlen - als Einnahme nieder, sondern hat auch indirekt Einfluß auf die Zahl der Zuschauer. (Wenn die Eintrittskarten zu teuer werden, kommen auch weniger Zuschauer)

3. Trainerwechsel

Insgesamt dreimal können Sie den Trainer wechseln. Ob aber dadurch die Spielstärke größer oder geringer wird, hängt vom Glück (oder anders ausgedrückt, von der RND-Funktion) ab.

Nicht beeinflussen können Sie:

1. Spielpaarungen

Wie schon erwähnt sind alle Spielpaarungen der 1. Fußball-Bundesliga für die Saison 85/86 als Daten im Programm enthalten. Da auch die Reihenfolge der Spieltage beachtet wurde, ergibt sich beim Spiel ein realistischer Saison-Durchlauf.

2. Formstärke

In welcher Tages-Form "Ihre" Mannschaft in ein Spiel geht, wird vom Programm berechnet. Hier kann es - wie in der Realität- ganz schöne Überraschungen geben.

3. Spielergebnisse

Natürlich können Sie die Ergebnisse nicht direkt eingeben; das wäre ja noch schöner. Das Spielergebnis "Ihrer" - und aller anderen - Mannschaften, wird durch sehr komplexe Berechnungen bestimmt. In die Berechnung gehen unter anderem die Spielstärke, die Formstärke, der Heimvorteil, der Tabellenplatz und das unberechenbare Glück ein.

4. Tabellenstand

Der Tabellenstand wird nach jedem Spieltag berechnet und richtet sich selbstverständlich nach den Ergebnissen aller bis dahin durchgeführten Spiele.

Der Spielablauf

Sie werden zuerst aufgefordert, aus den 18 Mannschaften der 1. Bundesliga den Verein zu wählen, den Sie managen möchten. Diese Festlegung ist für die Saison nicht mehr änderbar. Sie können also nicht, nachdem Sie einen Verein fast ruiniert haben, sich schnell ein neues "Opfer" aussuchen.

Es erscheinen dann alle Spielpaarungen für den 1. Spieltag. Durch Drücken einer beliebigen Taste gelangen Sie in das Hauptmenue.

Sie finden hier Ihren Gegner und die aktuellen Spiel- und Formstärken sowie ein Auswahlmenue, um die veränderbaren Größen zu beeinflussen oder mit Hilfe diverser Tabellen den Überblick zu wahren. Wenn Sie die Optionen des Menues nicht in Anspruch nehmen wollen, können Sie auch gleich mit "W" zum Spiel übergehen.

Sie gelangen auf die eigentliche Spielseite. Hier läuft Ihr erstes Spiel. Für die nächsten 90 Minuten (selbstverstänlich gerafft) sitzen Sie auf der harten Bank am Spielfeld und können nur noch hoffen und fluchen.

Nach dem Schlußpfiff können Sie sich auch die anderen Ergebnisse ansehen. Als letzte Seite folgt die Tabelle.

Nach erneutem Drücken einer Taste gelangen Sie wieder auf die Bildschirmseite, auf der die Paarungen des nunmehr zweiten Spieltages erscheinen.

Der Kreis ist nun geschlossen und Sie können - sicherlich mit wachsender Spannung - alle 34 Spieltage durchspielen.

Da Sie als Manager den großen Überblick behalten müssen, haben Sie noch die Möglichkeit (nach dem ersten Spieltag), vom Hauptmenue aus einen Rückblick anzuwählen. Hier können Sie die sportliche Entwicklung "Ihres" Vereins analysieren.

Fortsetzung von Seite 19

ReAss bei der Arbeit

Nach erfolgten Eingaben, erscheint die Meldung:

Programm   Start-Adresse - End-Adresse

Anschließend durchläuft ReAss den PASS 1 und den PASS 2. Erfolgt beim Ablegen des Source-Codes im Speicher ein Überlauf, wird der Inhalt auf dem entsprechenden Datenträger gesichert.

Wurde direkt aus dem Speicher reassembliert, erscheint bei:

Diskette: 'Disk einlegen'. (Was mit Return bestätigt wird)

Kassette: 'Press Record and Play'.

Denken Sie daran, daß im Kassetten-Betrieb nur ein File gerettet werden kann.

Zum Abschluß verabschiedet sich ReAss mit den Informationen über Programm-Anfang und -Ende und einer Meldung bei eventuellem Speicher-Überlauf. Der Rechner wird in den BASIC-Modus zurückgeschaltet.

Sie können mit der üblichen Sequenz SAVE"Name",8 ihr Source-File abspeichern, falls dies nicht gewählt wurde oder automatisch geschah. Mit LIST könen Sie sich das erzeugte Source-Listing ansehen.

SID-Kurs Teil 4

Die Programmierung mehrstimmiger Musikstücke ist Thema von Teil 4 der Serie über den Soundchip im C64. Die zu den unten abgedruckten Listings gehörigen Programme können Sie aus dem Kurs durch die übliche Tastenkombination CTRL und s abspeichern.

Geben Sie ab Zeile 500 eigene Notenwerte in DATA-Zeilen ein.

Listing 1

```
5 rem beispielprogramm 1 teil 4
10 forx=0to24:poke54272+x,0:nextx:poke54277,1*16+4:poke54278,6*16+10
20 poke54277+7,01*16+04:poke54278+7,6*16+10:poke54272+24,15
100 readl1,h1:poke54276,32:ifh1=0then120
110 poke54276,33:poke54272,l1:poke54273,h1
120 readl2,h2:poke54283,32:ifh2=0then140
130 poke54276+7,33:poke54272+7,l2:poke54273+7,h2
140 forw=1to200:nextw:ifl1<>-1then100
500 rem datas
510 data 180,8,180,8,-1,0,0,0
```

Listing 2

```
5 rem beispielprogramm 2 teil 4
10 forx=0to24:poke54272+x,0:nextx:poke54277,1*16+4:poke54278,6*16+10
20 poke54277+7,1*16+4:poke54278+7,6*16+10:poke54272+24,15
25 poke54277+14,1*16+4:poke54278+14,6*16+10
100 readl1,h1:poke54276,32:ifh1=0then120
110 poke54276,33:poke54272,l1:poke54273,h1
120 readl2,h2:poke54283,32:ifh2=0then133
130 poke54276+7,33:poke54272+7,l2:poke54273+7,h2
133 readl3,h3:poke54290,32:ifh3=0then140
135 poke54276+14,33:poke54272+14,l3:poke54273+14,h3
140 forw=1to200:nextw:ifl1<>-1then100
500 rem datas
510 data 0,8, 0,32, 0,128, -1,0, 0,0, 0,0
```

HINWEISE FÜR AUTOREN

Falls Sie uns ein Programm zur Veröffentlichung anbieten wollen, beachten Sie bitte folgende Hinweise:

Daß Ihre Programme lauffähig und absturzsicher sein müssen, versteht sich von selbst. Im einzelnen heißt das:

Kein Programmabbruch durch Fehlermeldungen, alle möglichen Eingabefehler werden abgefangen, die Bildschirmmaske wird nicht zerstört und so weiter.

Das Programm sollte in C-64-BASIC oder in 6502/6510-Assembler geschrie-ben sein. Als Hilfsmittel können die bisher in INPUT 64 erschienenen Tools (Hiresspeed, Sprite-Befehle usw.) benutzt werden. Sie können allerdings auch eine andere Hochsprache (zum Beispiel PASCAL) verwenden, für die es möglich ist, selbstständig lauffähige Compilate zu erzeugen. Näheres erfragen Sie bitte bei der Redaktion.

Ihr Programm sollte nicht länger als 100 Blöcke (25 KByte) sein.

Alle Programme müssen auch ohne Floppy lauffähig sein. Floppy-Betrieb optional ist erlaubt und gewünscht.

Senden Sie uns Ihre Programme bitte auf Kassette oder Diskette mit kommentiertem Listing und Beschreibung.

Sowohl Autostart als auch Listschutz erschweren uns nur die Arbeit! Wir werden deshalb Programme, deren Analyse absichtlichtlich erschwert wurde, zukünftig ungeprüft zurücksenden.

Wichtig: Sie müssen im Besitz der vollen Urheberrechte an Ihrem Programm sein und überlassen es uns zur Erstveröffentlichung.

Außerdem gibt es einige, durch das INPUT 64-Betriebssystem bedingte, programmiertechnische Erfordernisse:

1. Belegen Sie nur den Bereich des normalen BASIC-RAM ($0800-$9FFF) und unter dem BASIC-ROM ($A000-$BFFF).

2. Jede Benutzung von Zero-Page-Adressen, Veränderung der Betriebssystem-Vektoren (Interrupt, Tastatur, etc.) muß genau dokumentiert sein.

3. Die Programme müssen als BASIC-File zu laden und mit RUN zu starten sein.

4. Die CTRL-Taste darf nicht benutzt werden.

Und geben Sie bitte auf Listings, Kassetten, Disks und so weiter den Programmnamen sowie Ihre Anschrift an.

Hilfsprogramme: Sound-Erweiterung

Diese Tools ermöglichen dem BASIC-Programmierer eine einfache Nutzung der Sound-Möglichkeiten ohne umständliche POKE-Befehle.

Die Befehle:

Version 1 ($C800)	Version 2 ($0900)	
SYS 51200,vo	SYS 2304,vo	Lautstärke vo wird eingestellt.
SYS 51314,st,we(,pu)	SYS 2418,st,we(,pu)	Für Stimme st (0,1,2) wird Wellenform we (16,32,64,128 oder Mischformen) eingestellt. Ist 64 (Rechteck) enthalten, muss noch pu angegeben werden. Dies ist die Pulsbreite in Prozent.
SYS 51210,st,a,d,s,r	SYS 2314,st,a,d,s,r	Für Stimme st wird die Hüllkurve eingestellt.
SYS 51404	SYS 2508	Ein voreingestellter Ton wird gespielt. (BEEP)
SYS 51404,st,freq,da	SYS 2508,st,freq,da	Stimme st wird mit Frequenz freq gespielt. Nach da Interrupts (=1/60sek.) wird RELEASE eingeleitet. (PLAY)
SYS 51404,st,"note",da	SYS 2508,st,"note",da	dto., aber statt Frequenz kann eine Note angegeben werden.

Für A,D,S,R und VO können Werte von 0 bis 15 eingesetzt werden. Für die Pulsbreite (PU) sind Werte von 0 bis 99 zulässig, für DA werden Werte von 0 bis 255 akzeptiert.

Der Wert ST gibt die angesprochene Stimme des SID an, dabei werden Werte von 0 bis 2 ausgwertet.

Die Wellenformen entsprechen den üblichen Werten (16,32,64,128) oder Mischformen daraus. Die Frequenz (FREQ) beim PLAY-Befehl kann Werte von 0 bis 4095 annehmen.

Wird eine Note eingegeben, muß dies natürlich ein String sein; er kann eine maximale Länge von drei Zeichen haben. "C1" spielt das eingestrichene C, "C#4" spielt das Cis der vierten Oktave, die anderen Noten werden nach dem selben Format angegeben. Bei nicht zulässigen Angaben wird ein ILLEGAL DIRECT ERROR ausgegeben.

Dabei ist noch anzumerken, daß auch enharmonische Verwechslungen richtig gespielt werden.

Um eine gespielte Note zu unterbrechen, kann man POKE 51820+ST,1 bzw. POKE 2924+ST,1 eingeben, und beim nächsten IRQ wird die Release-Phase der Stimme eingeleitet.

Man kann auch überprüfen, ob diese schon eingeleitet ist, dazu sind die oben angeführten Adressen mit PEEK auszulesen und zu überprüfen, ob die Speicherstelle schon auf Null gesetzt war. Ist dies der Fall, befindet sich diese Stimme in der Release-Phase.

Um festzustellen, ob für alle Stimmen die Release-Phase eingeleitet wurde, kann man abfragen, ob Adresse 788 den Wert 49 beinhaltet. Wenn dies der Fall ist, stehen die Gate-Bits aller drei Stimmen auf Null.

Bitte beachten Sie, daß vor dem Aufruf der jeweiligen Stimme (SYS 2508,st,freq,da / SYS 51404,st,freq,da) die Parameter für Lautstärke, Wellenform und Hüllkurve durch die entsprechenden SYS-Aufrufe übergeben werden müssen. Dies gilt natürlich nicht für den voreingestellten Ton (BEEP).

Für die einwandfreie Funktion der Sound-Erweiterung darf der Interrupt-Vektor nicht für andere Zwecke verwendet werden, da dieser vom PLAY-Befehl intensiv genutzt wird.

Im Demo-Programm kann zwischen zwei Versionen ausgewählt werden. Die erste verschiebt nach einem RUN das Maschinenprogramm nach $C800 (verträglich mit Supertape) und löscht sich dann selbst. Das heißt, der BASIC-Speicherbereich wird nicht beeinflußt.

Die zweite Version legt nach dem Starten den BASIC-Anfang nach $0C00 (3072 dez.) und führt dann ein RUN aus. Man kann anschließend ein BASIC-Programm laden (mit Sekundäradresse 0) oder neu editieren. Vor dem Speichern müssen Sie POKE 44,8 eingeben, um die Sound-Tools mit auf Ihren Datenträger zu überspielen. Dann steht die Erweiterung, wenn das Programm später geladen und gestartet wird, sofort zur Verfügung und das BASIC-Programm wird abgearbeitet.

Listing-Sound-Erweiterung

```
]
0010
0020              ;SOUNDERWEITERUNG (V. 2.8.1985)
0030              ;VON SVEN HESSELBACH
0040
0050CHKKOM    .DE $AEFD        ;PRUEFT AUF KOMMA IM BASIC-TEXT
0060GETBYTE   .DE $B79E        ;HOLT WERT VON 0-255 INS X-REGISTER
0070ERROR     .DE $E38B        ;BASIC-WARMSTART/FEHLERAUSGABE
0080FACARG    .DE $BC0C        ;HOLT FAC NACH ARG
0090FACMALARG .DE $BA2B        ;MULTIPLIZIERT FAC MIT ARG
0100FACINT    .DE $B7F7        ;WANDELT FAC IN POSITIVEN INTEGER UM
0110INTFAC    .DE $B399        ;WANDELT POSITIVEN INTEGER IN FAC
0120INTFAC1   .DE $B395        ;WANDELT ZAHL IN Y/A IN FAC UM
0130FRMEVL    .DE $AD9E        ;WERTET EINEN AUSDRUCK IM BASICTEXT AUS
0140FRESTR    .DE $B6A6        ;HOLT STRINGZEIGER
0150
0160ADBY      .DE 679          ;ZWISCHENSPEICHER F. ENVELOPE
0170VOICE     .DE 680          ;BEINHALTET AKTUELLE STIMME
0180FREQLOW   .DE 681          ;SPEICHER FUER FREQUENZ
0190FREQHIGH  .DE 682
0200STRZEIGER .DE 683          ;HILFSZEIGER F. PLAY
0210
0220              ;VOLUME: SYS ST,STIMME,VOLUME
0230
0240VOLUME    JSR CHKKOM       ;AUF KOMMA PRUEFEN
0250          JSR GETBY        ;WERT VON 0-15 HOLEN
0260          STX 54296        ;IM LAUTSTAERKEREGISTER ABLEGEN
0270          RTS              ;ENDE
0280          ;----------------------------------
0290
0300              ;ENVELOPE: SYSST,STIMME,A,D,S,R
0310
0320ENVELOPE  JSR CHKKOM       ;AUF KOMMA PRUEFEN
0330          JSR GETVOICE     ;DIE AKRUELLE STIMME IN VOICE HOLEN
0340          JSR CHKKOM       ;AUF KOMMA PRUEFEN
0350          JSR GETBY        ;ATTACK HOLEN
0360          TXA              ;IN DEN AKKU UEBERTRAGEN
0370          ROL A            ;MIT 16 MULTIPLIZIEREN
0380          ROL A
0390          ROL A
0400          ROL A
0410          STA ADBY         ;UND ZWISCHENSPEICHERN
0420          JSR CHKKOM       ;AUF KOMMA PRUEFEN
0430          JSR GETBY        ;DECAY HOLEN
0440          TXA              ;IN DEN AKKU UEBERTRAGEN
0450          ORA ADBY         ;MIT ATTACK VERKNUEPFEN
0460          LDX VOICE        ;INDEX FUER SID HOLEN (0,7,14)
0470          STA 54277,X      ;ATTACK UND DECAY ABSPEICHERN
0480          JSR CHKKOM       ;AUF KOMMA PRUEFEN
0490          JSR GETBY        ;SUSTAIN HOLEN
0500          TXA              ;IN DEN AKKU UEBERTRAGEN
0510          ROL A            ;MIT 16 MULTIPLIZIEREN
0520          ROL A
0530          ROL A
0540          ROL A
0550          STA ADBY         ;UND ZWISCHENSPEICHERN
0560          JSR CHKKOM       ;AUF KOMMA PRUEFEN
0570          JSR GETBY        ;RELEASE HOLEN
0580          TXA              ;IN DEN AKKU UEBERTRAGEN
0590          ORA ADBY         ;MIT SUSTAIN VERKNUEPFEN
0600          LDX VOICE        ;INDEX FUER SID HOLEN (0/7/14)
0610          STA 54278,X      ;UND ABSPEICHERN
0620          RTS              ;ENDE
0630          ;----------------------------------
0640GETBY     JSR GETBYTE      ;EIN BYTE INS X-REGISTER HOLEN
0650          CPX #16          ;GROESSER ALS 15 ?
0660          BCS ILQU         ;JA, DANN ILLEGAL QUANTITY
0670          RTS              ;SONST ZURUECK
0680          ;----------------------------------
0690ILQU      LDX #14          ;ILLEGAL QUANTITY ERROR
0700          JMP ERROR        ;AUSGEBEN UND 'READY.'
0710          ;----------------------------------
0720GETVOICE  JSR GETBYTE      ;EIN BYTE AUS BASICTEXT HOLEN
0730          CPX #3           ;GROESSER ALS 2 ?
0740          BCS ILQU         ;JA, DANN ILLEGAL QUANTITY
0750          CPX #0           ;GLEICH NULL ?
0760          BEQ VOICE0       ;JA, DANN X NICHT VERAENDERN
0770          CPX #2           ;GLEICH ZWEI ?
0780          BEQ STI2         ;JA, DANN X-REGISTER AUF 14 SETZEN
0790          LDX #7           ;SONST X-TREGISTER AUF 7 SETZEN (STIMME 2)
0800          .BY 44           ;NAECHSTEN BEFEHL UEBERGEHEN
0810STI2      LDX #14          ;STIMME 3
0820VOICE0    STX VOICE        ;STIMME (0/7/14) ABSPEICHERN
0830          RTS              ;ENDE
0840          ;----------------------------------

0850
0860          ;WAVE: SYS ST,WAVE(,PULSE)
0870
0880WAVE      JSR CHKKOM       ;AUF KOMMA PRUEFEN
0890          JSR GETVOICE     ;STIMME HOLEN
0900          JSR CHKKOM       ;AUF KOMMA PRUEFEN
0910          JSR GETBYTE      ;EIN BYTE HOLEN
0920          TXA              ;IN DEN AKKU UEBERTRAGEN
0930          PHA              ;UND AUF STACK RETTEN
0940          AND #64          ;IST RECHTECK IN DER WELLE ENTHALTEN ?
0950          BNE RECWAVE      ;JA, DANN ZU RECWAVE (PULSE HOLEN)
0960RUECKRECW PLA              ;WELLENFORM VOM STACK HOLEN
0970          AND #240         ;ALLES ANDERE AUSBLENDEN
0980          LDX VOICE        ;STIMME HOLEN
0990          BEQ STIMME0      ;UND AUS 0/7/14  ZAHLEN 0/1/2 MACHEN
1000          CPX #7
1010          BNE NICHTSTI1
1020          LDX #1
1030          .BY 44
1040NICHTSTI1 LDX #2
1050STIMME0   STA WELLE,X      ;WELLENFORM IN ZWISCHENSPEICHER ABLEGEN
1060          RTS              ;ENDE
1070
1080RECWAVE   JSR CHKKOM       ;AUF KOMMA PRUEFEN
1090          JSR GETBYTE      ;PULSE IN PROZENT HOLEN
1100          CPX #100         ;GROESSER ALS 99 ?
1110          BCC PULSEOK      ;NEIN, DANN OK,
1120          JMP ILQU         ;SONST ILLEGAL QUANTITY
1130PULSEOK   STX *$63         ;X-REGISTER IN $63 ABLEGEN
1140          LDA #0
1150          STA *$62         ;HIGH BYTE GLEICH NULL
1160          JSR INTFAC       ;POSITIVEN INTEGER IN FAC HOLEN
1170          LDX #5           ;KONSTANTE1 IN ARG UEBERTRAGEN
1180ARGLOOP   LDA CONST1,X
1190          STA *105,X
1200          DEX
1210          BPL ARGLOOP
1220          JSR FACMALARG    ;FAC MIT ARG (40.96) MULTIPLIZIEREN
1230          JSR FACINT       ;WIEDER IN POSITIVEN INTEGER UMWANDELN
1240          LDX VOICE        ;0/7/14 HOLEN
1250          STA 54275,X      ;HIGH BYTE V. PULSE ABSPEICHERN
1260          TYA              ;LOW-BYTE UEBERTRAGEN
1270          STA 54274,X      ;UND ABSPEICHERN
1280          JMP RUECKRECW    ;ZURUECK INS 'WAVE'-PROGRAMM
1290          ;----------------------------------
1300
```

```
1310           ;PLAY: SYS ST,"NOTE",LAENGE
1320           ;ODER: SYS ST,FREQUENZ,LAENGE
1330           ;ODER: SYS ST
1340
1350PLAY       JSR $0079          ;LETZES ZEICHEN HOLEN
1360           BNE PLAYNORM       ;UNGLEICH NULL? DANN PLAY-BEFEHLE
1370           JMP PRESET         ;SONST ZUM 'BEEP'
1380
1390PLAYNORM   JSR CHKKOM         ;AUF KOMMA PRUEFEN
1400           JSR GETVOICE       ;STIMME HOLEN
1410           JSR CHKKOM         ;AUF KOMMA PRUEFEN
1420           JSR FRMEVL         ;AUSDRUCK AUSWERTEN
1430           BIT *$0D           ;TYPFLAG ABFRAGEN
1440           BPL FREQU          ;NUMERISCH? DANN FREQUENZ AUSWERTEN
1450           JMP NOTE1          ;SONST ALS NOTE AUSWERTEN
1460
1470FREQU      JSR FACINT         ;FAC IN POS. INTEGER UMWANDELN
1480FREQESP    CMP #16            ;FREQUENZ ZU GROSS?
1490           BCC FREQUOK        ;NEIN, DANN WEITER
1500           JMP ILQU           ;SONST ILLEGAL QUANTITY
1510
1520FREQUOK    SEC                ;UMWANDLUNG VORBEREITEN
1530           JSR INTFAC1        ;WANDELT INTEGER IN Y/A IN FAC UM
1540           JSR FACARG         ;FAC NACH ARG UEBERTRAGEN
1550           LDX #5             ;KONSTANTE2 NACH FAC HOLEN (17.02....)
1560ARGLOOP2   LDA CONST2,X
1570           STA *97,X
1580           DEX
1590           BPL ARGLOOP2
1600           JSR FACMALARG      ;FAC MIT ARG MULTIPLIZIEREN
1610           LDA *102           ;VORZEICHEN IMMER POSITIV
1620           AND #127
1630           STA *102
1640           JSR FACINT         ;FAC IN POSITIVEN INTEGER UMWANDELN
1650           LDX VOICE          ;0/7/14 FUER STIMME HOLEN
1660           STA 54273,X        ;HIGH-BYTE ABSPEICHERN
1670           TYA                ;LOW-BYTE UEBERTRAGEN
1680           STA 54272,X        ;UND ABSPEICHERN
1690FREQSP     JSR CHKKOM         ;AUF KOMMA PRUEFEN
1700           JSR GETBYTE        ;BYTE (LAENGE) HOLEN
1710           SEI                ;STOERUNG DURCH IRQ VERBIETEN
1720           TXA                ;X-REGISTER IN AKKU UEBERTRAGEN
1730           LDX VOICE          ;STIMME HOLEN UND IN 0/1/2 UMWANDELN
1740           BEQ STIMME01
1750           CPX #7
1760           BNE NOTVCO1
1770           LDX #1
1780           .BY 44
1790NOTVCO1    LDX #2
1800STIMME01   STA NOTE,X         ;LAENGE IN NOTE (+1/2) ABSPEICHERN
1810           LDA #1             ;GATE-BIT
1820           ORA WELLE,X        ;MIT WELLENFORM VERKNUEPFEN
1830           LDX VOICE          ;INDEX HOLEN
1840           STA 54276,X        ;UND WELLE+GATE ABSPEICHERN
1850           LDA #L,IRQESP      ;INTERRUPT UMLEGEN

2410           LDA ($22),Y        ;IN DEN AKKU LADEN
2420           CMP #'#'           ;IST ES EIN KREUZ ?
2430           BEQ KREUZ          ;JA, DANN SPRUNG
2440           LDX #6             ;7 NOTENBUCHSTABEN VERGLEICHEN
2450           LDY #0             ;MIT DEM ERSTEN ZEICHEN IM STRING
2460SUNOTE     LDA NOTEN,X        ;NOTE HOLEN
2470           CMP ($22),Y        ;MIT 1. ZEICHEN IM STRING VERGLEICHEN
2480           BEQ NOTEFD         ;GLEICH, DANN NOTE GEFUNDEN
2490           DEX                ;SONST X ERNIEDRIGEN
2500           BPL SUNOTE IN WEITERSUCHEN
2510ILLDIR     LDX #21            ;WENN TABELLE DURCH UND NICHT GEFUNDEN
2520           JMP ERROR          ;ILLEGAL DIRECT ERROR AUSGEBEN
2530
2540NOTEFD     TXA                ;NOTENNUMMER IN DEN AKKU UEBERTRAGEN
2550           ASL A              ;UND MIT ZWEI MULTIPLIZIEREN
2560           TAX                ;UND WIEDER IN DEN AKKU UEBERTRAGEN
2570           LDA FREQTAB,X      ;FREQUENZ LOW LADEN
2580           STA FREQLOW        ;ABSPEICHERN
2590           INX
2600           LDA FREQTAB,X      ;FREQUENZ HIGH LADEN
2610           STA FREQHIGH       ;ABSPEICHERN
2620FREQTEILNG INY                ;LETZTES ZEICHEN IM STRING
2630           LDA ($22),Y        ;IN AKKU LADEN
2640           SEC                ;AUS ASCII-WERT ZAHL MACHEN
2650           SBC #$30
2660           CMP #8             ;GROESSER ALS 7 ?
2670           BCC OKTAVOK        ;JA, DANN SPRUNG
2680           JMP ILQU           ;NEIN, DANN ILLEGAL QUANTITY
2690OKTAVOK    STA STRZEIGER      ;OKTAVE ABSPEICHERN
2700           CMP #7             ;FREQUENZ MULTIPLIZIEREN ?
2710           BNE FREQTEI        ;NEIN, DANN SPRUNG
2720           ASL FREQLOW        ;FREQUENZ MAL ZWEI
2730           ROL FREQHIGH
2740           JMP FREQOK         ;UN ZU FREQUOK
2750
2760FREQTEI    LDX #7             ;AUSGANGSWERT: OKTAVE 6
2770FREQERNLP  DEX                ;ERNIEDRIGEN
2780           CPX STRZEIGER      ;VERGLEICHEN MIT ANGEGEBENER OKTAVE
2790           BEQ FREQOK         ;GLEICH, DANN ZU FREQUOK
2800           LSR FREQHIGH       ;SONST FREQUENZ HALBIEREN
2810           ROR FREQLOW
2820           JMP FREQERNLP      ;VORGANG WIEDERHOLEN
2830
2840FREQOK     LDA FREQHIGH       ;FREQUENZ IN Y/A HOLEN
2850           LDY FREQLOW
2860           JMP FREQESP        ;EINSPRUNG BEI FREQESP
2870
2880KREUZ      LDX #6             ;NOTE SUCHEN (S.O.)
2890           LDY #0
2900SUKRZN     LDA NOTEN,X
2910           CMP ($22),Y
2920           BEQ KREUZNFD
2930           DEX
2940           BPL SUKRZN
2950           JMP ILLDIR
```

```
1860            STA 788
1870            LDA #H,IRQESP
1880            STA 789
1890            CLI                 ;INTERRUPT ERLAUBEN
1900            RTS                 ;ENDE
1910IRQESP      LDX #2              ;INDEX IST ZWEI
1920            LDY #0              ;ZAEHLER FUER ABGESCHALTETE STIMMEN IST 0
1930            STY YREG            ;WENN ZAEHLER = 3, DANN IRQ ZURUECK (S.U.)
1940IRQLOOP1    LDA NOTE,X          ;ZAEHLER HOLEN
1950            BNE SPRNO02         ;NICHT NULL, DANN ERNIEDRIGEN
1960            INC YREG            ;SONST ZAEHLER ERHOEHEN
1970            BNE SPRNO01         ;UNBEDINGTER SPRUNG
1980
1990SPRNO02     DEC NOTE,X          ;ZAEHLER ERNIEDRIGEN
2000            BNE SPRNO01         ;NICHT NULL ? DANN SPRUNG
2010            LDA WELLE,X         ;SONST STIMME ABSCHALTEN
2020            PHA
2030            TXA
2040            PHA
2050            CPX #0
2060            BEQ SPRSTI0
2070            CPX #1
2080            BEQ SPRSTI2
2090            LDX #14
2100            .BY 44
2110SPRSTI2     LDX #7
2120SPRSTI0     PLA
2130            TAY
2140            PLA
2150            STA 54276,X
2160            TYA
2170            TAX
2180SPRNO01     DEX                 ;ANGESPROCHENE STIMME ERNIEDRIGEN
2190            BPL IRQLOOP1        ;ZAHL POSITIV ? DANN SCHLEIFE
2200            LDY YREG            ;ANZAHL AUSGESCHALTETER STIMMEN
2210            CPY #3              ;GLEICH DREI ?
2220            BNE IRQEND          ;NEIN, ZUM INTERRUPT
2230            LDA #$31            ;SONST IRQ-VEKTOR ZURUECKLEGEN
2240            STA 788
2250            LDA #$EA
2260            STA 789
2270IRQEND      JMP $EA31           ;ENDE DER IRQ-ROUTINE
2280
2290STRTOLONG   LDX #23             ;STRING TOO LONG ERROR
2300            JMP ERROR           ;AUSGEBEN UND PGM ABBRECHEN
2310            ;-------------------------------
2320            ;PROGRAMMTEIL FUER NOTE AUSWERTEN
2330NOTE1       JSR FRESTR          ;ZEIGER AUF NOTE UND LAENGE DES STRINGS
2340            CMP #4              ;HOLEN--LAENGER ALS 3 ?
2350            BCS STRTOLONG       ;DANN STRING TOO LONG
2360            CMP #2              ;SONST VERGLEICH, OB KLEINER ALS 2
2370            BCS NOTEOK          ;NEIN, DANN OK
2380            JMP STRTOLONG       ;FEHLERMELDUNG
2390
2400NOTEOK      LDY #1              ;ZWEITES ZEICHEN IM STRING

2960KREUZNFD    TXA                 ;NOTE GEFUNDEN, FREQUENZ AUS
2970            ASL A               ;KREUZTAB HOLEN
2980            TAX
2990            LDA KREUZTAB,X
3000            BEQ ILLDIR
3010            STA FREQLOW
3020            INX
3030            LDA KREUZTAB,X
3040            STA FREQHIGH
3050            LDY #1
3060            JMP FREQTEILNG      ;UND ENTSPRECHEND TEILEN
3070            ;-------------------------------
3080PRESET      LDA #0              ;VOREINSTELLUNG SPIELEN
3090            STA 54272+11
3100            STA 54272+18
3110            LDA #15
3120            STA 54272+24
3130            STA 54272+6
3140            STA 54272
3150            STA 54272+5
3160            LDA #70
3170            STA 54272+1
3180            LDA #33
3190            STA 54272+4
3200            LDX #160
3210            LDY #255
3220PRESETLP    DEY
3230            BNE PRESETLP
3240            DEX
3250            BNE PRESETLP
3260            LDA #0
3270            STA 54272+1
3280            STA 54272+24
3290            STA 54272+4
3300            STA 54272+5
3310            STA 54272+6
3320            RTS
3330            ;-------------------------------
3340WELLE       .BY 32 32 32 ;ZWISCHENSPEICHER FUER WELLENFORM
3350NOTE        .BY 0 0 0 ;ZAEHLER FUER SUSTAIN-PHASE
3360CONST1      .BY 134 163 215 10 62 0 ;KONSTANTE 40.96 F. PULSE
3370CONST2      .BY 133 136 58 45 110 0 ;KONST. 17.02... F. FREQUENZ->PARAMETER
3380YREG        .BY 0               ;ZAEHLER FUER STIMMEN (IRQ)
3390NOTEN       .BY 'CDEFGAH'       ;MOEGLICHE NOTEN
3400FREQTAB     .BY 23 4 151 4 ;FREQUENZEN DER NOTEN IN OKTAVE 6
3410            .BY 39 5 117 5
3420            .BY 32 6 224 6
3430            .BY 184 7
3440KREUZTAB    .BY 85 4 220 4 ;FREQUENZEN DER NOTEN MIT KREUZ
3450            .BY 117 5 200 5
3460            .BY 125 6 73 7
3470            .BY 8 45
//
]
```

Für Nachzügler und "Spätzünder"

Wegen der großen Nachfrage haben wir bereits vergriffene Ausgaben von INPUT 64 nachproduziert, so daß alle bisher erschienenen Ausgaben wieder lieferbar sind!! Ab Ausgabe 4/85 ist INPUT 64 auch auf Diskette erhältlich. Richten Sie Ihre Bestellung bitte direkt an den Verlag; die Lieferung erfolgt nur gegen Vorkasse (Verrechnungsscheck).
Kassettenversion: DM 12,80;
Diskettenversion: DM 19,80.
(incl. Porto und Verpackung)



INPUT 64
Vertriebsabteilung
Verlag Heinz Heise GmbH
Postfach 2746

3000 Hannover 1

ÜBRIGENS: Abonnenten sparen den Preis von zwei INPUT 64 - Ausgaben pro Jahr (Diskettenversion). Und zwar ohne den üblichen "Pferdefuß", denn das INPUT 64 - Abo ist jederzeit mit Wirkung ab der übernächsten Ausgabe kündbar!

Aus dem Inhalt: Ausgabe 1 - Dateiverwaltung, drei (!) Spiele * Ausgabe 2 - Textprogramm, Zeichengenerator * Ausgabe 3 - Spriteeditor, Maschinensprache-Monitor * Ausgabe 4 - SuperTape DII, Grafikhilfe, Urlaubskalender * Ausgabe 5 - Mathe mit Nico, Talk to me (Dialogsimulation), Hintergrundmonitor * Ausgabe 6 - Haushaltsbuchführung, Recorder-Justage, Textadventure, BASIC-Compactor * Ausgabe 7 - Scroll Editor, TapeCopy, HiFi-Boxenberechnung.

Rätselecke

Wir freuen uns immer wieder über die Vielzahl der eingesandten Lösungen. Auch diesmal haben viele Leser einen Lösungsvorschlag zu unserer Aufgabe eingesandt, ein vorgegebenes Datenfeld möglichst schnell zu sortieren.

Den Hauptpreis hat ein Leser aus dem deutschsprachigen Ausland gewonnen:

Christof Mader aus Graz in Österreich

Alle Gewinner werden von uns schriftlich benachrichtigt.

Bei der Auswertung dieses Rätsels haben es uns die Einsender sehr schwer gemacht, die ersten zehn Gewinner aus den Einsendungen herauszufinden.

Einige Programme waren zwar etwas schneller als das von Christof Mader (sie brauchten zum Sortieren so um die fünf bis fünfzehn Sekunden), aber entweder wurden zusätzliche Felder definiert oder benutzt - einige Programmierer übersahen einfach, daß die Feld-Variable F$(0) noch zur Benutzung frei war - oder es wurde nur nach dem ersten Buchstaben sortiert, wie zum Beispiel hier:

APPS V.
ALBRECHT P.

Diese beiden Namen müßten selbstverständlich getauscht werden.

In manchen Fällen wurde auch das Ende-Kennzeichen (*) nicht berücksichtigt und so weiter und so weiter.

Rätsellösung

```
300 AN=I-1:S=INT(AN/1.39)
310 FORI=1TOAN-S
320 IFF$(I)>F$(I+S)THENH$=F$(I+S):F$(I+S)=F$(I):F$(I)=H$
330 NEXT:IFS<>2THENS=INT(S/1.39):GOTO310
340 FORI=2TOAN:IFF$(I)>=F$(I-1)THEN390
350 H$=F$(I):FORE=I-1TO1STEP-1:F$(E+1)=F$(E):IFH$<=F$(E-1)THEN380
370 F$(E)=H$:GOTO390
380 NEXTE
390 NEXTI
```

Oder so:

Platz zwei - entwickelt von Peter Firmenich

```
300 N=I-1:A=49152:B=49302:C=1:D=N:I=1
310 E=C:F=D
320 IFF$(E)<=F$(F)THEN420
330 Z$=F$(E):F$(E)=F$(F):F$(F)=Z$
360 E=E+1:IFE=FTHEN440
380 IFF$(E)<=F$(F)THEN360
390 Z$=F$(E):F$(E)=F$(F):F$(F)=Z$
420 F=F-1:IFF<>ETHEN320
440 F=F+1:IFF>=DTHEN480
460 POKEA+I,F:POKEB+I,D:I=I+1
480 E=E-1:IFC>=ETHEN520
500 D=E:GOTO310
520 I=I-1:C=PEEK(A+I):D=PEEK(B+I):IFI>0THEN310
```

Platz drei - entwickelt von Bernd Leps

```
300 N=I-1:S=820:B=20:POKES,1:POKE840,N:L=1:R=N
340 I=L:J=R:A$=F$((L+R)/2)
360 IFA$<F$(J)THENJ=J-1:GOTO360
370 FORI=ITON:IFF$(I)<A$THENNEXT
380 IFI>JTHEN450
390 H$=F$(I):F$(I)=F$(J):F$(J)=H$:I=I+1:J=J-1:IFI<=JTHEN360
450 IFI<RTHENS=S+1:POKES,I:POKEB+S,R
460 R=J:IFL<RTHEN340
470 IFS=820THEN998
490 L=PEEK(S):R=PEEK(B+S):S=S-1:IFL<RTHEN340
520 IFS-820THEN490
```

Wenn man sich diese drei Programm-Listings ansieht, merkt man, daß sie gar nicht direkt miteinander verglichen werden können.

Jeder der drei Erstplazierten hat das Problem auf eine andere Art und Weise gelöst.

Alle drei Programme könnten aber noch so verändert werden, daß sie noch schneller laufen. Zum Beispiel durch Ausschalten des Bildschirmes (Bit vier in Register 53265 löschen) oder Ausschalten des Interrupts (Bit

eins im Register 56334 löschen). Wobei zu beachten ist, daß nach dem Programm-Durchlauf der alte Wert im Register 56334 wieder herzustellen ist.

Viele Leser haben nicht nur die Lösung zum Rätsel eingeschickt, sondern auch einen Brief dazu geschrieben. Hier sind zwei Kostproben:

...Die Beschäftigung mit dem gestellten Problem hat Freude gemacht, auch falls vielleicht nichts dabei herauskommt.

J. Rätsch

...es ist eine gute Idee, statt eines Rätsels eine Programmaufgabe zu stellen, da anzunehmen ist, daß in jedem INPUT 64-Bezieher noch soviel Energie und Freude an der Sache steckt, daß eine gestellte Aufgabe als persönliche Herausforderung angesehen und deren Lösung sofort in Angriff genommen wird.

In meinem Falle war das ebenso und dabei habe ich auch noch neue Erkenntnisse gewonnen und Spass gehabt.

R. Strauch

Diesen Meinungen haben wir uns angeschlossen. Wir stellen Ihnen deshalb eine neue Aufgabe, die auch wieder aus einer Mischung von Rätsel und Programmierung besteht.

Jetzt geht es los!

Wir kommen zu unserer neuen Aufgabe. Im Mittelpunkt steht dabei der Satz: Send more Money

Übersetzt lautet er: Schick mehr Geld. Formatiert man den Satz etwas anders, dann kann man daraus eine tolle Knobelei machen. Für diese drei Wörter, aus denen der Satz besteht, sollen drei Zahlen eingesetzt werden, wobei jeder Buchstabe eine Ziffer darstellt.

```
   SEND
 + MORE
 ------
  MONEY
```

Stellen Sie sich vor, daß jeder Buchstabe eine Ziffer darstellt. Die Wörter 'SEND', 'MORE' und 'MONEY' entsprechen dann drei verschiedenen Zahlen. Wenn Sie die Zahlen-Codes der Wörter 'SEND' und 'MORE' addiert haben, sollten Sie die entsprechende Zahlen-Kombination des Wortes 'MONEY' erhalten.

Das von Ihnen zu erstellende Programm sollte in der Lage sein, diese und ähnliche Aufgaben lösen zu können. Auch dürfte das Programm nicht nur für jeden Buchstaben alle Ziffern durchprobieren (das kann ja jeder), sondern sich durch eigene "Intelligenz" auszeichnen.

Außerdem möchten wir von Ihnen einen ähnlichen, sinnvollen Spruch wie oben haben, der ebenfalls von Ihrem Programm enträtselt werden muß.

Für das letzte Rätsel verlangten wir ausschließlich eine reine BASIC-Lösung. Diesmal erweitern wir den Sprachschatz und lassen noch eine weitere Programmier-Sprache zu, nämlich PASCAL. Für die Lösungen in PASCAL sind die gleichen Preise ausgesetzt wie für die BASIC-Lösung.

Das von Ihnen eingesandte Programm soll nach dem Starten die gestellte Aufgabe in möglichst kurzer Zeit lösen und zum Schluß die drei Zahlen untereinander stehend auf dem Bildschirm anzeigen.

Wenn Sie an diesem Rätsel teilnehmen wollen, schicken Sie uns Ihre Lösung als BASIC- oder PASCAL-Listing zu. Bitte notieren Sie auch auf dem Lösungsblatt die Zeit, die Ihr Programm zum Erstellen der Lösung benötigt (Bei PASCAL-Programmen den verwendeten PASCAL-Compiler).

Der Einsendeschluß für Ihr Listing ist der 1. Oktober 1985 (Datum des Poststempels). Der Rechtsweg ist wie immer ausgeschlossen.

Die Anschrift entnehmen Sie bitte dem Impressum.

Gewinnen können Sie selbstverständlich auch.

Der Autor, der das schnellste Programm zur Lösung der gestellten Aufgabe entwickelt hat, erhält ein Jahresabonnement INPUT 64 (wahlweise Kassetten- oder Diskettenversion). Außerdem warten wieder neun Buchpreise auf einen Empfänger.

Wir wünschen Ihnen viel Erfolg beim Entwickeln des Programmes. Bitte vergessen Sie nicht, auf dem Listing die Laufzeit des Programmes anzugeben.

BEI LADEPROBLEMEN:

Schimpfen Sie nicht auf uns, die Bänder sind normgerecht nach dem neuesten technischen Stand aufgezeichnet und sorgfältig geprüft.

Sondern: Reinigen Sie zunächst Tonköpfe und Bandführung Ihres Kassettenrecorders. Sie können dazu eine Reinigungskassette verwenden, gründlicher und besser ist es aber, ein Wattestäbchen und Reinigungsflüssigkeit zu verwenden. Die genaue Vorgehensweise ist im Handbuch der Datasette beschrieben.

Führt auch dies nicht zum Erfolg, ist wahrscheinlich der Tonkopf Ihres Gerätes verstellt. Dieser Fehler tritt leider auch bei fabrikneuen Geräten auf.

TONKOPF SELBST JUSTIEREN

Wir haben ein Programm entwickelt, mit dessen Hilfe Sie den Aufnahme-/Wiedergabekopf justieren können.

Tippen Sie das Programm JUSTAGE ein, und speichern Sie es ab. Dieses Programm wertet ein etwa 30 Sekunden langes Synchronisationssignal aus, das sich am Ende des Bandes befindet. Gehen Sie bitte folgendermaßen vor:

Nehmen Sie sich einen kleinen Schraubenzieher und werfen Sie einen Blick auf Ihre Datasette. Über der REWIND-Taste, in etwa 0,5 cm Abstand vom Kassettenfach, befindet sich ein kleines Loch. Wenn Sie die PLAY-Taste drücken und durch dieses Loch schauen, sehen Sie den Kopf der Justierschraube für die Spurlage.

Legen Sie nun die zweite Seite von INPUT-64 ein, und spulen Sie zum Bandanfang. Drücken Sie jetzt die PLAY-Taste, lassen Sie das Band 45 Sekunden laufen, dann stoppen und umdrehen. Das Band steht jetzt kurz vor dem Synchro-Signal.

Starten Sie das JUSTAGE-Programm mit RUN, jetzt sollte die Meldung "PRESS PLAY ON TAPE" kommen, drücken Sie also die PLAY-Taste. Nach dem Drücken der Taste geht der Bildschirm zunächst wie immer aus. Wird das Synchro-Signal erreicht, wechselt die Bildschirmfarbe; und zwar - bei nicht total verstellter Spurlage - völlig regelmäßig etwa dreimal pro Sekunde. Liegt die Spur des Tonkopfes grob außerhalb der zulässigen Toleranzgrenzen, geschieht entweder nichts oder die Farben wechseln unregelmäßig.

Geschieht dies nicht, dann verdrehen Sie die oben beschriebene Einstellschraube. Markieren Sie sich vorher die alte Stellung der Schraube, sonst kann es bei grob verstelltem Tonkopf passieren, daß Sie mit dejustiertem Tonkopf geschriebene Kassetten nicht mehr lesen können. Aber

Vorsicht: ganz langsam drehen, ohne dabei Druck auszuüben! Verdrehen Sie die Schraube nicht mehr als eine Umdrehung in jede Richtung. Nach etwas Ausprobieren wird der Bildschirm gleichmäßig die Farbe wechseln. Soweit die Grobeinstellung.

Zur Feineinstellung lassen Sie das Synchro-Signal noch einmal von Anfang an laufen. Die Schraube jetzt nach links drehen, bis der Farbwechsel unregelmäßig wird. Diese Stellung genau merken (am besten markieren), und die Schraube jetzt langsam wieder nach rechts drehen: Der Farbwechsel wird zunächst gleichmäßig, bei weiterem Drehen wieder unregelmäßig. Merken Sie sich auch diese Stellung, und drehen Sie die Schraube nun in Mittelstellung, das heißt zwischen die beiden Randstellungen. Denken Sie daran, daß während der Einstellung kein Druck auf den Schraubenkopf ausgeübt werden darf!

Der Tonkopf Ihres Recorders ist jetzt exakt justiert. Sollte sich auch nach dieser Einstellung INPUT 64 nicht laden lassen, erhalten Sie von uns eine Ersatzkassette. Schicken Sie dazu bitte die defekte Kassette mit einem entsprechenen Vermerk an den Verlag ein (Adresse siehe Impressum).

Besitzer der Ausgabe 6/85 können es sich einfacher machen. Das dort veröffentlichte Programm RECORDER-JUSTAGE macht die Einstellung des Daten-Recorders zum Kinderspiel.

Listing justage

```
800 fori=49199to49410:readd:ps=ps+d:pokei,d:next
900 ifps<>24716thenprint"falsch abgetippt - fehler korrigieren!":end
950 print"o.k."
970 sys12*16↑3+11*16+10
1000 rem von 49199 bis 49410
1010 data173, 13,220,169,217.174,  4,220,172,  5,220,141, 14,220, 48, 44, 56
1020 data102, 88, 36, 89, 48, 12,144, 10,165, 88,133, 90,169,128,133, 88,133
1030 data 91,192,121,144,  4,224,115,176,  7,169,  0,133, 92, 56,176, 11,165
1040 data 92, 73,128,133, 92, 36, 92, 16, 19, 24,102, 88, 36, 89, 48, 12,144
1050 data 10,165, 88,133, 90,169,128,133, 88,133, 91,104,168,104,170,104, 64
1060 data 96, 36, 91, 16,252,132, 91,165, 90, 96,160,128,132, 89,165, 88.201
1070 data 22,208,250,132, 88,160, 10,132, 89,132, 91, 36, 91, 16,252,132, 91
1080 data165, 90,201, 22,208,226,136,208,241, 32,133,192,201, 22,240,249, 96
1090 data 32,147,252,120, 32, 23,248,165,  1, 41, 31,133,  1,133,192,169, 47
1100 data141, 20,  3,169,192,141, 21,  3,169,127,141, 13,220,169,144,141, 13
1110 data220,173, 17,208, 41,239,141, 17,208,169, 70,141,  4,220,169,129,141
1120 data  5,220, 88, 32,142,192,201, 42,208,249,173, 32,208, 41, 15,168,200
1130 data140, 32,208, 76,237,192,208, 76

ready.
```

Ab 21. Oktober an Ihrem Kiosk: INPUT 64 Ausgabe 10/85

Wir bringen unter anderem:

— JETFLIGHT

Geschrieben von einem Berufspiloten: ein Flugsimulator mit den Daten der wichtigsten westdeutschen Flughäfen. Simuliert wird das Verhalten einer großen Verkehrsmaschine (DC 10, Boeing 747) im Instrumentenflug. Fast eine Übung für den Pilotenschein!

— ReLoc

Nachdem wir in dieser Ausgabe einen Re-Assembler vorgestellt haben, folgt gleich ein Relocator. Dieses Programm verschiebt Maschinenprogramme im Speicher und führt die notwendigen Änderungen durch. So ist zum Beispiel die Anpassung von SuperTape an jeden beliebigen Adressbereich möglich!

— und außerdem:

Hilfsprogramme * Serien: SID-Kurs, Mathe mit Nico, 64er-Tips * News * Spiele * und so weiter.

---

c't-Magazin für Computertechnik

c't 10/85 - jetzt am Kiosk

Prüfstand: Commodore 128, Schneider CPC6128, Robotext-Modul für C64, Operator II * Serien: Einsteigen in CP/M, Des Schneiders Kern, Compiler, Dr. Osborne Kit * Applikation: V-Chips von NEC ersetzen 8080 und 8086 * Software-Know-how: Einführung in Prolog * Projekte: Universelle Busmonitore für Z80 und 68000, Unicard, I/O-Karte für Apple u.v.a.m

elrad-Magazin für Elektronik

Heft 10/85 - ab 30.9. am Kiosk

* Technische Daten komplett: Modularer Vorverstärker 4 * Universal-Steuerplatine für Synthesizer * Elektronisches Kunstwerk: Perpetuum Pendulum * VCA mit OTA: Spannungsgesteuerter Verstärker * Wie kommt das Bild aufs Band: Video-Aufzeichnung * u.v.a.m

(Änderungen vorbehalten)

IMPRESSUM

INPUT 64
Das elektronische Magazin

Verlag Heinz Heise GmbH
Bissendorfer Str. 8
3000 Hannover 61
Postanschrift:
Postfach 610407
3000 Hannover 61
Tel.: (05 11) 53 52-0

Technische Anfragen
nur dienstags von 9-16.30 Uhr

Postgiroamt Hannover, Konto-Nr. 93 05-308
(BLZ 250 100 30)
Kreissparkasse Hannover, Konto-Nr. 000-01 99 68
(BLZ 250 502 99)

Herausgeber: Christian Heise

Redaktion:
Christian Persson (Chefredakteur)
Ralph Hülsenbusch
Wolfgang Möhle
Karl-Friedrich Probst
Jürgen Seeger

Ständige Mitarbeiter:
Peter S. Berk
Irene Heinen
Peter Sager
Hajo Schulz
Peter Seeliger
Eckart Steffens

Vertrieb: Anita Kreutzer

Redaktion, Anzeigenverwaltung, Abonnementsverwaltung:
Verlag Heinz Heise GmbH
Postfach 610407
3000 Hannover 61
Tel.: (05 11) 53 52-0

Grafische Gestaltung:
Wolfgang Ulber, Dirk Wollschläger

Herstellung: Heiner Niens

Lithografie:
Köhler & Lippmann, Braunschweig.

Druck:
Leunisman GmbH, Hannover
Hahn-Druckerei, Hannover

Konfektionierung:
Lettershop Brendler, Hannover

Kassettenherstellung:
SONOPRESS GMBH, Gütersloh

INPUT 64 erscheint monatlich.
Einzelpreis DM 12,80
Jahresabonnement Inland Kassette DM 140.-
Diskette DM 198,-
Diskettenversion im Direktbezug: DM 16,80
+ DM 3,- Porto und Verpackung

Vertrieb (auch für Österreich, Niederlande, Luxemburg und Schweiz):
Verlagsunion Zeitschriften-Vertrieb
Postfach 5707
D-6200 Wiesbaden
Ruf (0 61 21) 2 66-0

Verantwortlich:
Christian Persson
Bissendorfer Str. 8
3000 Hannover 61

ISSN 0177-3771

Titelidee: **INPUT 64**
Titelfoto: Artreferenz

Titelmusik: **INPUT 64**

Programmierung: H. Gehrmann

hier abtrennen

INPUT 64-Abonnement Abruf-Coupon

Ja, übersenden Sie mir bis auf Widerruf alle künftigen INPUT64-Ausgaben ab Monat

(Kündigung ist jederzeit mit Wirkung ab der jeweils übernächsten Ausgabe möglich. Überzahlte Abonnementsgebühren werden sofort anteilig erstattet.)

Das Jahresabonnement kostet: ○ auf Kassette DM 140,— inkl. Versandkosten und MwSt.

○ auf Diskette DM 198,— inkl. Versandkosten und MwSt.

(Bitte ankreuzen/Nichtzutreffendes streichen.)

Absender und Lieferanschrift

Bitte in jedes Feld nur einen Druckbuchstaben (ä = ae, ö = oe, ü = ue)

Vorname/Zuname

Beruf/Funktion

Straße/Nr.

PLZ Wohnort

Datum/Unterschrift

Von meinem Recht zum schriftlichen Widerruf dieser Order innerhalb einer Woche habe ich Kenntnis genommen. Zur Wahrung der Frist genügt die rechtzeitige Absendung.

Unterschrift

Bitte beachten Sie, daß diese Bestellung nur dann bearbeitet werden kann, wenn beide Unterschriften eingetragen sind.

INPUT 64-Abonnement Abruf-Coupon

Ich wünsche Abbuchung der Abonnement-Gebühr von meinem nachstehenden Konto. Die Ermächtigung zum Einzug erteile ich hiermit.

Name des Kontoinhabers

Bankleitzahl Konto-Nr.

Geldinstitut Ort des Geldinstituts

Bankeinzug kann nur innerhalb Deutschlands und nur von einem Giro- oder Postscheckkonto erfolgen.

Heise

Bitte im (Fenster-)Briefumschlag einsenden.
Nicht als Postkarte verwenden!

INPUT64

Vertriebsabteilung
Verlag Heinz Heise GmbH
Postfach 61 04 07

3000 Hannover 61